Information Leaks in Structured Peer-to-Peer Anonymous Communication Systems

PRATEEK MITTAL and NIKITA BORISOV, University of Illinois at Urbana-Champaign

We analyze information leaks in the lookup mechanisms of structured peer-to-peer (P2P) anonymous communication systems and how these leaks can be used to compromise anonymity. We show that the techniques used to combat active attacks on the lookup mechanism dramatically increase information leaks and the efficacy of passive attacks, resulting in a tradeoff between robustness to active and passive attacks.

We study this tradeoff in two P2P anonymous systems: Salsa and AP3. In both cases, we find that, by combining both passive and active attacks, anonymity can be compromised much more effectively than previously thought, rendering these systems insecure for most proposed uses. Our results hold even if security parameters are changed or other improvements to the systems are considered. Our study, therefore, shows the importance of considering these attacks in P2P anonymous communication.

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

General Terms: Security

Additional Key Words and Phrases: Anonymity, attacks, information leaks, peer-to-peer

ACM Reference Format:
Mittal, P. and Borisov, N. 2012. Information leaks in structured peer-to-peer anonymous communication systems. ACM Trans. Inf. Syst. Secur. 15, 1, Article 5 (March 2012), 28 pages.
DOI = 10.1145/2133375.2133380 http://doi.acm.org/10.1145/2133375.2133380

1. INTRODUCTION

Anonymous communication hides the identity of communication partners from third parties or hides user identity from the remote party. The Tor network [Dingledine et al. 2004], deployed in 2003, now serves hundreds of thousands of users and carries terabytes of traffic per day [The Tor Project]. Originally an experimental network used by privacy enthusiasts, it is now entering mainstream use; for example, several consulates use it to evade observation by their host country [Goodin 2007; Zetter 2010].

The capacity of Tor is already strained, and to support a growing population, a peer-to-peer approach will likely be necessary, as P2P networks allow the network capacity to scale with the number of users. Indeed, several proposals for peer-to-peer anonymous communication have been put forward [Freedman and Morris 2002; McLachlan et al. 2009; Mislove et al. 2004; Mittal and Borisov 2009; Nambiar and Wright 2006; Rennhard and Plattner 2002]. However, P2P networks present new challenges to anonymity, one of which is the ability to locate relays for anonymous traffic.

This material is based on work supported by the National Science Foundation under grants 0627671, 0831488, and 0953655.
Authors' address: P. Mittal and N. Borisov, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 W. Main Street, Urbana, IL 61801; email: {mittal2, nikita}@illinois.edu.

DOI 10.1145/2133375.2133380 http://doi.acm.org/10.1145/2133375.2133380

In Tor, clients use a directory to retrieve a list of all the running routers. Such a directory will not scale as the number of routers grows, since the traffic to update the directory would become prohibitively expensive [McLachlan et al. 2009]. Instead, a peer-to-peer lookup is needed to locate an appropriate relay. Such a lookup, however, can be subject to attack: malicious nodes can misdirect it to find relays that are colluding and violate the anonymity of the entire system. All of the P2P anonymous communication designs therefore incorporate some defense against such attacks; for example, AP3 [Mislove et al. 2004] uses secure routing techniques developed by Castro et al. [2002], and Salsa uses redundant routing with bounds checks [Nambiar and Wright 2006].

These defenses, however, come at a cost. They operate by performing extra checks to detect incorrect results returned by malicious nodes. These checks cause many messages to be exchanged between nodes in the network, some of which might be observed by attackers. As a result, a relatively small fraction of attackers can make observations about a large fraction of lookups that occur in the P2P network, acting as a near-global passive adversary. Modern anonymity networks are not designed to resist a global passive adversary, because such an attack is believed to be too difficult to mount for all but the most powerful adversaries, and because defenses against a global passive adversary are too costly for most users. Therefore, this small fraction of attackers can successfully attack anonymity of the system.

We examine this problem through a case study of two P2P anonymous communication systems: Salsa and AP3. In both systems, defenses against active attacks create new opportunities for passive attacks. Salsa and AP3 make heavy use of redundancy to address active attacks, rendering them vulnerable to passive information-leak attacks. Further, increasing the levels of redundancy will improve passive attack performance and will often make the system weaker overall. We find that even in the best case, Salsa is much less secure than previously considered. Salsa was designed to tolerate up to 20% of compromised nodes; however, our analysis shows that, in this case, over one quarter of all circuits will be compromised by using information leaks. Similarly, conventional analysis of AP3 suggests that it provides probable innocence when up to 33% of nodes are compromised and can tolerate up to 50% of compromised nodes by increasing the path length. However, our analysis puts these numbers at 5% and 10%, respectively.

We studied potential improvements to Salsa that can be achieved by increasing the path length or introducing a public key infrastructure (PKI). We found that these tools offer only a limited defense against our attacks, and the system is still not secure for practical purposes. Our results demonstrate that information leaks are an important part of anonymity analysis of a system.

The article is organized as follows. In Section 2 we present the state of low-latency anonymous communication. We discuss information leaks from lookups in Section 3 and show the trade off between security and anonymity. In Sections 4 and 5, we present attacks based on information leaks from lookups on AP3 and Salsa. In Section 6, we present an entropy-based approach to computing-information leaks in Salsa. Section 7 contains related work, and we conclude in Section 8.

2. BACKGROUND

In this section, we present a brief overview of anonymous communication. We motivate the need for decentralized and scalable solutions and discuss why structured peer-to-peer systems have strong potential. We also describe our threat model.

2.1. Low-Latency Anonymous Communication Systems

Anonymous communication systems can be classified into low-latency and high-latency systems. High-latency anonymous communication systems like Mixminion [Danezis et al. 2003] and Mixmaster [Möller et al. 2003] are designed to be secure even against a powerful global passive adversary; however, the message transmission times for such systems are typically on the order of several hours. This makes them unsuitable for use in applications involving interactive traffic like Web browsing and instant messaging. The focus of this article is on low-latency anonymous communication systems [Boucher et al. 2000; Clarke et al. 2001; Dingledine et al. 2004; I2P 2003].

Tor [Dingledine et al. 2004] is a popular low-latency anonymous communication system. Users (clients) download a list of servers from central directory authorities and build anonymous paths using onion routing [Syverson et al. 2000]. There are several problems with Tor's architecture. First, the reliance on central directory authorities makes them an attractive target for the attackers. Second, Tor serves hundreds of thousands of users, and the use of a relatively small number of servers to build anonymous paths becomes a performance bottleneck. Finally, Tor requires all users to maintain a global view of all the servers. As the number of servers increases, maintaining a global view of the system becomes costly, since churn will cause frequent updates and a large bandwidth overhead. In order to address these problems, a peer-to-peer architecture will likely be necessary. However, peer-to-peer networks present new challenges to anonymity, one of which is the ability to locate relays for anonymous traffic.

Several designs for peer-to-peer low-latency anonymous communication have been proposed. Tarzan [Freedman and Morris 2002] replaced the centralized directory authority with a gossip protocol that was used to distribute knowledge of all peers to all other peers. While decentralized, the requirement that each node maintain an up-to-date global view of the system means that the system could scale only to about 10,000 nodes. MorphMix [Rennhard and Plattner 2002] was designed to scale to much larger network sizes. It built an unstructured peer-to-peer overlay between all the relays and created paths along this overlay to forward anonymous communications. Nodes along the path are queried for their neighbors in order to choose the next hop. To prevent a node from providing malicious results, a scheme using witness nodes and a collusion detection mechanism is used. However, the collusion detection mechanism can be circumvented by a set of colluding adversaries who model the internal state of each node, thus violating anonymity guarantees [Tabriz and Borisov 2006].

Several other designs have used so-called structured peer-to-peer topologies [Mislove et al. 2004; Nambiar and Wright 2006], also known as distributed hash tables (DHTs), as a foundation for anonymous peer-to-peer communication. Structured topologies assign neighbor relationships using a pseudorandom but deterministic mathematical formula based on the IP addresses or public keys of nodes. This allows the relationships to be verified externally, presenting fewer opportunities for attacks. AP3 [Mislove et al. 2004] used a secure lookup mechanism [Castro et al. 2002] in the Pastry DHT [Rowstron and Druschel 2001] to select random forwarders and used them to build an anonymous communication path. The secure lookup techniques are based on a PKI and, thus, do not achieve a truly decentralized security model. The lookup was also not designed to be anonymous, a property that we will show to have important consequences for the security of AP3.

Salsa [Nambiar and Wright 2006] aimed to offer secure P2P anonymous communication in a system without a PKI. It designed a custom DHT structure and a custom secure lookup mechanism specifically tailored for the purposes of anonymous

communication. Its secure lookup and path construction mechanisms rely heavily on redundancy to detect potential attacks. As we will show, such redundancy creates information leaks, and presents a trade-off between resisting active attacks and presenting more opportunities for passive attacks.

2.2. Threat Model

Low-latency anonymous communication systems are not designed to to be secure against a global passive adversary. We consider a partial adversary who controls a fraction f of all the nodes in the network. This set of malicious nodes colludes and can launch both passive and active attacks. We consider the set of colluding nodes static, and the adversary cannot compromise nodes at will. In terms of the standard terminology introduced by Raymond [2000], our adversary is internal, active, and static.

Even in networks with large numbers of nodes, f can be a significant fraction of the network size. Both Salsa and AP3 use mechanisms to prevent Sybil attacks [Douceur 2002], which would otherwise allow an adversary to attain an f arbitrarily close to 1. However, powerful adversaries, such as governments or large organizations, can potentially deploy enough nodes to gain a significant fraction of the network. Similarly, botnets, whose size often measures in tens to hundreds of thousands of nodes [Cooke et al. 2005; Rajab et al. 2006; Holz et al. 2008], present a very real threat to anonymity. In this work, we consider values of f up to 0.2.

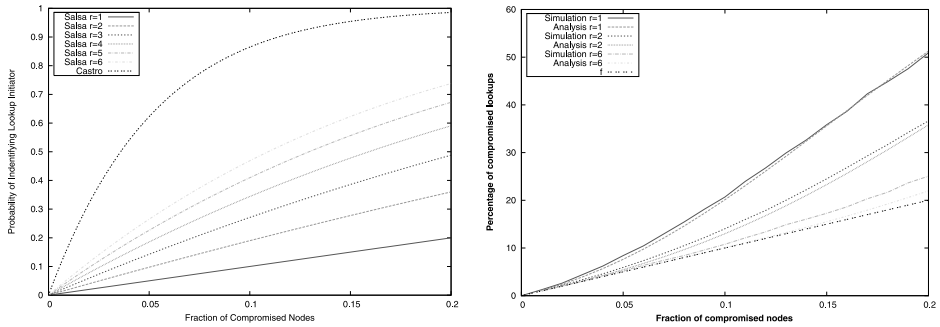
3. INFORMATION LEAKS VIA SECURE LOOKUPS

It has been recognized that unprotected DHTs are extremely vulnerable to attacks on the lookup mechanism. First, malicious nodes can perform a Sybil attack [Douceur 2002] and join the network many times, increasing the fraction f. Second, they can intercept lookup requests and return incorrect results by listing a colluding malicious node as the closest node to a key, thus increasing the fraction of lookups that return malicious nodes. Finally, they can interfere with the routing table maintenance and cause the routing tables of honest nodes to contain a larger fraction of malicious nodes; this will increase the chance that a lookup can be intercepted and the result can be subverted.

3.1. Castro et al.'s Secure Lookup

Castro et al. [2002] designed a suite of mechanisms to counter these attacks. We discuss their mechanisms in context of Pastry [Rowstron and Druschel 2001], a structured peer-to-peer overlay network, though they are applicable to other DHTs. They proposed the following.

— *Secure node identifier assignment.* Each node is issued a certificate by a trusted authority, which binds the node identifier with a public key. The authority limits the number of certificates and prevents Sybil attacks.
— *Secure routing table maintenance.* Even with secure nodeID assignment, attackers can maliciously influence routing table construction. The Pastry routing algorithms allow flexibility in selecting a neighbor for each slot, which is used for optimizing latency or other metrics. Attackers can exploit this flexibility by suggesting malicious choices for these slots. Secure routing table maintenance eliminates this flexibility by creating a parallel, constrained routing table where each slot can have only a single possible node, as verified by secure lookup. This solution ensures that, on average, only a fraction f of a node's neighbors will be malicious.
— *Secure lookups (secure message forwarding).* For secure lookups, a two-phase approach is employed. The message is routed via the normal routing table (optimized



(a) Informatıon leak from secure lookups. (b) Percentage of compromised lookups.

Fig. 1. Salsa lookup mechanism.

for latency), and a routing failure test is applied. If the test detects a failure, redundant routing is used, and all messages are forwarded according to the constrained routing table. The failure test makes use of the observation that the density of honest nodes is greater than the density of malicious nodes. The idea behind redundant routing is to ensure that multiple copies of messages are sent to the key root via diverse routes. Note that Castro et al. [2002] consider the problem of securely routing to the entire replica set, for which a neighbor anycast mechanism is also used.

Used together, these techniques are quite effective at ensuring that a lookup returns the actual closest node to the randomly chosen identifier, which in turn suggests that it is malicious with probability f. However, the secure lookup mechanism generates many extra messages: the routing failure test involves contacting the entire root set of a node (L immediate neighbors in the nodeID space), and redundant routing sends a request across several paths. These messages let attackers detect when a lookup has been performed between two honest nodes with high probability. The probability of detecting the lookup initiator can be approximated as $1 - (1 - f)^{L+\lceil \log_{2^b} N \rceil - 1}$, which is quite high for the typical values of $L = 16$ and $b = 4$. In Figure 1(a), we plot the probability of detection of the lookup initiator as a function of the fraction of compromised nodes f using $N = 1{,}000$. We can see that a small fraction of 5% of compromised nodes can detect the lookup initiator more than 60% of the time. Moreover, when the fraction of compromised nodes is about 10%, the lookup initiator is revealed 90% of the time.

This shows the fundamental tension that is encountered by a DHT lookup. The default Pastry mechanisms provide little defense against active adversaries who try to disrupt the lookup process, dramatically increasing the probability that a lookup returns a compromised node. Castro et al.'s mechanisms solve this problem but introduce another, as the lookup is no longer anonymous and can be observed by malicious nodes. A relatively small fraction of malicious nodes can, therefore, act as a near-global passive adversary and compromise the security of anonymous communication systems. The secure lookup exposes nodes to increased surveillance; we note that this may have consequences for protocols other than anonymous communication that are built on top of secure lookup.

3.2. Salsa Secure Lookup

Salsa [Nambiar and Wright 2006] is based on a custom DHT that maps nodes to a point in an ID space corresponding to the hash of their IP address. The ID space in Salsa is divided into groups and organized into a binary tree structure. Each node

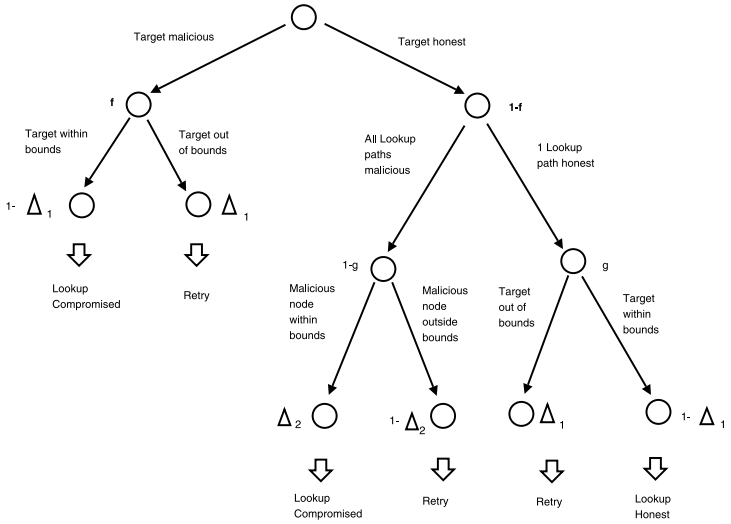


Fig. 2. Computing probability of a compromised lookup.

knows all the nodes in its group (local contacts) and a small number of nodes in other groups (global contacts).

Similar to Pastry, nodes must rely on other nodes to perform a recursive lookup. A malicious node that intercepts the request could return the identity of a collaborating attacker node. Salsa makes use of redundant routing and bounds checks to reduce the lookup bias. The Salsa architecture is designed to ensure that redundant paths have very few common nodes between them (unlike Pastry or Chord [Stoica et al. 2003]). This reduces the likelihood that a few nodes will be able to modify the results for all the redundant requests. A lookup initiator asks r local contacts (chosen at random) to perform a lookup for a random key. The returned value that is closest to the key is selected, and a bounds check is performed. If the distance between the prospective owner and the key is greater than a threshold distance b, it is rejected, reasoning once again that malicious nodes are less dense than honest ones and, thus, will fail the bounds check much more frequently. If the bounds check test fails, the result of the lookup is discarded, and another lookup for a new random key is performed. Redundant routing and the bounds check work together: an attacker would need to both intercept all of the redundant lookups and have a malicious node that is close enough to avoid the bounds check.

We first perform an analysis of the security of the Salsa lookup protocol. Let us denote the initiator of the lookup by I and the target identifier by ID. We have to consider two possibilities: either the (actual) successor of ID is honest, or it is malicious (see Figure 2). For a random ID, the probability that a node is malicious will be f. The next question is whether this malicious node will pass the bounds check; let us call Δ_1 the probability that it fails. In the case of failure, the current lookup is aborted, and a new one is initiated. If the test is passed, the malicious node is returned as the result of the lookup.

If the successor of ID is honest, on the other hand, the lookup will return that honest node if there is at least one lookup path composed of only honest nodes. Let us say that this happens with probability g. In this case, the honest node must still pass the bounds check to obtain an honest result of the lookup; so the lookup is aborted with probability Δ_1. (Note that this is the same regardless of whether the successor of ID

is honest or malicious, since ID was picked uniformly at random in each case). If, on the other hand, every path has a malicious node (with probability $1-g$), the malicious nodes can suggest the malicious node closest to ID as the result. This malicious node will also be subject to the bounds check. It is more likely to fail the test now, because it is no longer the closest node to ID; so let us call the probability of failure Δ_2. A failed bounds check will cause the lookup to be restarted. A successful check will result in the malicious node being returned.

Δ_1 is the probability of false positives during a bounds check; that is, there is no node with an identifier in the range between target ID and $ID + b$, where b is the bounds check parameter. If we consider the ID space to be the interval $[0, 1)$, then Δ_1 can be computed as

$$\Delta_1 = (1-b)^N. \tag{1}$$

Δ_2 is the probability of a false negative; that is, given that the target node is honest, there is a malicious node within bounds. Suppose that the target node is at a distance a from ID. The cumulative density function (CDF) of this distance is given by $F(a) = (1-a)^N$, and the PDF is given by $f(a) = N \cdot (1-a)^{N-1}$. Now, we have

$$\begin{aligned}
\Delta_2 &= P(\text{malicious node within bounds}|\ \text{target node is honest}), && \text{(2a)}\\
&= 1 - P(\text{malicious node outside bounds}|\text{target node is honest}), && \text{(2b)}\\
&= 1 - \int_{a=0}^{b} f(a) \cdot \left(\frac{1-b}{1-a}\right)^{N \cdot f} da - \int_{a=b}^{1} f(a) \cdot 1\, da, && \text{(2c)}\\
&= 1 - \int_{a=0}^{b} N \cdot (1-a)^{N-1} \cdot \left(\frac{1-b}{1-a}\right)^{N \cdot f} da - \int_{a=b}^{1} N \cdot (1-a)^{N-1} da, && \text{(2d)}\\
&= 1 - N \cdot (1-b)^{N \cdot f} \cdot \frac{1-(1-b)^{N-N \cdot f}}{N - N \cdot f} - \Delta_1. && \text{(2e)}
\end{aligned}$$

g is the probability that there is at least one lookup path with all honest nodes. This probability depends on the lookup path lengths. For simplicity, let us first consider the case of a single lookup ($r = 1$). We shall later extend our analysis for redundant lookups.

3.2.1. Single Lookup, $r = 1$. Let us denote the lookup path length by L. Given a particular lookup path length ($L = l$), we have

$$g_l = P(\text{Lookup is honest}|L = l) = (1-f)^l. \tag{3}$$

Based on Figure 2, we have

$$P(\text{Compromised Lookup}|L = l) = \frac{f \cdot (1-\Delta_1) + (1-f) \cdot (1-g_l) \cdot \Delta_2}{f \cdot (1-\Delta_1) + (1-f) \cdot (1-g_l) \cdot \Delta_2 + (1-f) \cdot g_l \cdot (1-\Delta_1)}, \tag{4}$$

where Δ_1, Δ_2, and g_l have been computed in Equations (1), (2), and (3). Note that we need to factor out aborted lookups, since we are interested in the fraction of successful lookups that produce a malicious node.

Now we shall compute $P(L = l)$. Let D denote the distance between the initiator I's group and target ID's group, in terms of the number of levels of the binary tree structure. This is illustrated in Figure 3. In order to compute $P(L = l)$, we can first condition the event $D = d$. Since I selects the target ID uniformly at random from

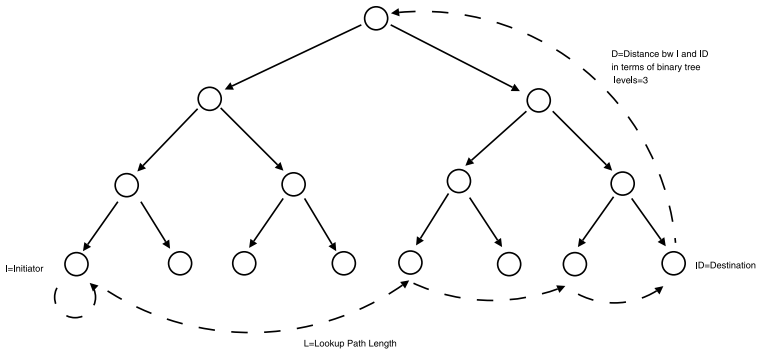


Fig. 3. Salsa binary tree structure.

the ID space, the probability that the target is d levels away from the initiator in the binary tree structure is

$$P(D = d) = \begin{cases} \frac{2^{d-1}}{|G|} & d \geq 1 \\ \frac{1}{|G|} & d = 0 \end{cases}, \tag{5}$$

where $|G|$ is the number of groups in Salsa.

Under the event $D = d$, we shall compute the probability of lookup path length being l hops, that is, $P(L = l|D = d)$. The lookup from I to ID can proceed along several different paths, depending on local contact chosen by the initiator. Note that the first hop is always a local contact in the initiators group, and the last hop is always in the target group. Thus we need to select $l - 2$ more hops from among the $d - 1$ possible *subgroup levels* relative to the target ID. Subgroup level refers to a set of nodes who have the same binary tree distance (levels) from the target ID. The probability of selecting any subgroup level is 1/2. Thus, given $D = d$, the total number of possible lookup paths of length l is $\binom{d-1}{l-2}$, where the probability of selecting any individual path is $(\frac{1}{2})^{d-1}$. From the above, we have

$$P(L = l|D = d) = \begin{cases} \binom{d-1}{l-2}\left(\frac{1}{2}\right)^{d-1} & d \geq 1 \\ 1 & d = 0, l = 1 \\ 0 & d = 0, l > 1 \end{cases}. \tag{6}$$

Using Equations (5) and (6), we can compute $P(L = l)$ as follows.

$$P(L = l) = \sum_{d=0}^{\log_2 |G|} P(L = l|D = d) \cdot P(D = d), \tag{7a}$$

$$P(L = l) = \begin{cases} \sum_{d=1}^{\log_2 |G|} \binom{d-1}{l-2} \cdot \frac{1}{|G|} & l \geq 2 \\ \frac{1}{|G|} & l = 1 \end{cases}. \tag{7b}$$

Finally, using Equations (4) and (7) we can compute the probability of a compromised lookup as

$$P(\text{Compromised Lookup}) = \sum_{l=1}^{(\log_2 |G|)+1} P(\text{Compromised Lookup}|L = l) \cdot P(L = l). \tag{8}$$

3.2.2. Redundant Lookups. Let us denote the r lookup path lengths by $L_1, \ldots, L_r$. Given particular lookup path lengths ($L_1 = l_1, \ldots, L_r = l_r$), we have

$$g = P(\text{at least one lookup path is honest}), \tag{9a}$$
$$= 1 - P(\text{all lookup paths have a malicious node}), \tag{9b}$$
$$= 1 - \prod_{j=1}^{r} \left(1 - (1-f)^{l_j}\right). \tag{9c}$$

Based on Figure 2, we have

$$P(\text{Compromised Lookup}|L_1 = l_1, \ldots, L_r = l_r) = \frac{f \cdot (1-\Delta_1) + (1-f)\cdot(1-g)\cdot\Delta_2}{f \cdot (1-\Delta_1) + (1-f)\cdot(1-g)\cdot\Delta_2 + (1-f)\cdot g \cdot (1-\Delta_1)}, \tag{10}$$

where Δ_1, Δ_2, and g have been computed in Equations (1), (2), and (9). Now we shall compute $P(L_1 = l_1, \ldots, L_r = l_r)$ by conditioning on the event $D = d$. Note that conditioned on $D = d$, the redundant lookups are independent. Thus, we have

$$P(L_1 = l_1, \ldots, L_r = l_r|D = d) = \prod_{j=1}^{r} P(L_j = l_j|D = d). \tag{11}$$

Using Equation (11), we can compute $P(L_1 = l_1, \ldots, L_r = l_r)$ as

$$P(L_1 = l_1, \ldots, L_r = l_r) = \sum_{d=0}^{\log_2|G|} P(L_1 = l_1, \ldots, L_r = l_r|D = d) \cdot P(D = d), \tag{12a}$$
$$= \sum_{d=0}^{\log_2|G|} \left(\prod_{j=1}^{r} P(L_j = l_j|D = d)\right) \cdot P(D = d), \tag{12b}$$

where $P(L = l|D = d)$ and $P(D = d)$ are given by Equations (6) and (5). Finally, using Equations (10) and (12), we can compute the probability of a compromised lookup as

$$P(\text{Compromised Lookup}) = \sum_{l_1=1}^{(\log_2|G|)+1} \cdots \sum_{l_r=1}^{(\log_2|G|)+1} P(\text{Compromised Lookup}|L_1 = l_1, \ldots, L_r = l_r) \cdot P(L_1 = l_1, \ldots, L_r = l_r). \tag{13}$$

To validate our mathematical model, we used a simulator developed by the authors of Salsa [Nambiar and Wright 2007].[1] The simulator was configured to simulate 1,000 topologies, and in each topology, results were averaged over 1,000 random lookups. The lookup bias is sensitive to the average lookup path length, which in turn is sensitive about $\log_2|G|$, where $|G|$ is the number of groups. This is because longer path lengths give attackers more opportunities to intercept the lookup and subvert the result. We therefore used 128 groups, which would be a typical number in a large network, and 1,000 nodes in our simulation. Salsa is resistant to conventional attacks that target the lookup mechanism as long as the fraction of malicious nodes in the system is less than 20%. Since Salsa does not provide adequate security for higher values of f, we

[1]Our results differ slightly from those shown in Nambiar and Wright [2006] because of a bug in the original simulator that we fixed. We have communicated the bug to the authors who have confirmed it.

shall limit our analysis to $f \leq 0.2$. In Figure 1(b), we study the effect of varying redundancy on the lookup bias. The curve $y = f$ is shown as a reference for an optimal secure lookup protocol. Note that the simulation results closely match our analytic calculations. We can also see that increasing r clearly reduces the fraction of compromised lookups, thus increasing security. For $f = 0.2$, the fraction of compromised lookups drops from 37% to 24% when r is increased from 2 to 6. The initiator of a lookup can be identified by the attackers if any of the local contacts used for redundant lookups are compromised. The probability of detecting the lookup initiator is $1-(1-f)^r$, as depicted in Figure 1(a). Clearly, increasing r increases the chance that a lookup initiator is detected. This illustrates the tradeoff between security and anonymity of a lookup.

In this section, we observed that secure lookups leak information about the lookup initiator. Furthermore, we observed a tradeoff between the security and anonymity of a lookup. A relatively small fraction of malicious nodes are able to observe a large fraction of lookups. Next, we will use this to break the anonymity of AP3 and Salsa.

4. AP3

AP3 [Mislove et al. 2004] is an anonymous communication system built on top of Pastry [Rowstron and Druschel 2001]. The essence of AP3 operation is similar to Crowds [Reiter and Rubin 1998], where a random walk over all of the nodes in the system is used to forward requests while concealing the initiator's identity. In both AP3 and Crowds, a node A that wants to send a message to a node B first picks a random relay F_1 to forward the message. F_1 then flips a weighted coin, and with probability p, it chooses another relay, F_2, and forwards the request there. With probability $1-p$, F_1 delivers the message directly to the recipient B.

Therefore, a message is forwarded through a path of nodes, all of which are selected randomly. The path length follows a geometric distribution, with the expected length being $\frac{1}{1-p}$. We can assume that some of the relays will be malicious and will try to guess the identity of the initiator. However, due to the stochastic nature of the forwarding, such relays will have a hard time telling whether they received a message from the initiator directly, or from another relay. Reiter and Rubin first analyzed the probability that the initiator is correctly identified [1998]; we review the terminology used in their analysis here, as we will extend it in later sections.

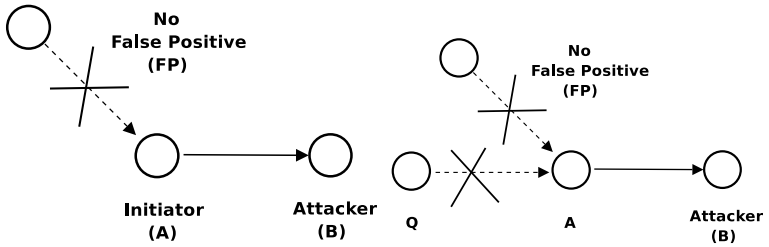
Let H_k denote the event that the first attacker in the forwarding path occupies the kth position, where the initiator is at the 0th. Let $H_{k+} = H_k \vee H_{k+1} \vee H_{k+2} \vee \ldots$ and let I denote the event that attackers identified the initiator correctly (as the predecessor). Then, given that an attacker intercepts a message, the chance that the initiator guessed correctly is $P(I|H_{1+})$. This can be further decomposed as

$$P(I|H_{1+}) = \frac{P(I \wedge H_{1+})}{P(H_{1+})} = \frac{P(H_1)P(I|H_1) + P(H_{2+})P(I|H_{2+})}{P(H_{1+})}. \quad (14)$$

Note that $P(I|H_1) = 1$, since in this case the initiator is identified correctly, and $P(I|H_{2+}) = 0$. If we let f represent the fraction of nodes that are compromised, then

$$P(I|H_{1+}) = \frac{P(H_1)}{P(H_{1+})} = \frac{f}{\sum_{i=1}^{\infty} \left(p(1-f)\right)^{i-1} f}.$$

Reiter and Rubin proposed the notion of *probable innocence* as happening whenever the true initiator is identified with a probability less than 1/2. By solving $P(I|H_{1+}) < 1/2$ for f, we can see that as long as $f < 1 - \frac{1}{2p}$, probable innocence will be assured. For example, with $p = 0.75$, up to 33% nodes can be malicious without compromising probable innocence. By increasing p, even larger fractions of compromised nodes can



(a) $E_1 \wedge H_1$. (b) $E_1 \wedge H_{2+}$.

Fig. 4. Information leak in AP3.

be tolerated, up to the limit of 50%, when $p = 1$. (Of course, larger p results in longer paths.)

4.1. The E_1 Attack

The chief difference between AP3 and Crowds is the manner in which the relays are chosen. Both aim to pick a relay at random out of all the nodes in the system, but Crowds assumes that all nodes know about all other nodes, which does not scale. AP3 uses the secure lookup due to Castro et al. to locate relays. To pick a relay, a node performs a secure lookup in the Pastry DHT for a random key. This, in turn, can be used to break probable innocence. In addition to the base observation—node A used malicious node B as a relay—the malicious nodes have an extra observation point: whether any other node has performed a lookup for node A. We will define the event E_1 as the case when no lookups for A have been detected. (E_1 implies H_{1+}.) We can then calculate the probability $P(I|E_1)$, such that

$$P(I|E_1) = \frac{P(I \wedge E_1)}{P(E_1)}.$$

To calculate $P(E_1)$, we need to consider two cases: either A is, in fact, the initiator (H_1), or some other node, Q, forwarded the request to A (H_{2+}). In the former case, E_1 will be true unless there is another spurious lookup (false positive) for A, due to another request that is detected by the attackers. We call the spurious lookup event FP. In the latter scenario, we need two things to happen: first, no spurious lookup has happened, and second, the lookup from Q to A was not detected. We call this second event Q. Figure 4 represents the analysis of the two cases.

Therefore, we can express E_1 as

$$E_1 \equiv (H_1 \wedge \neg FP) \vee (H_{2+} \wedge Q \wedge \neg FP)$$

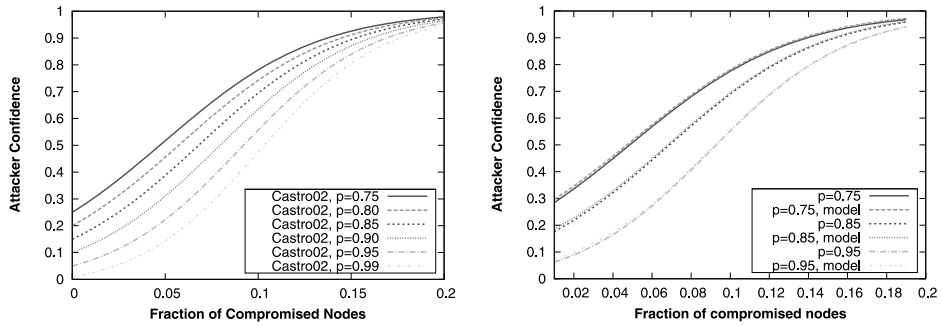
Because H_1 and H_{2+} are mutually exclusive, and FP and Q are independent from H_1, H_{2+}, and each other, we can write

$$P(E_1) = P(H_1)P(\neg FP) + P(H_{2+})P(\neg FP)P(Q).$$

Therefore,

$$\begin{aligned} P(I|E_1) &= \frac{P(H_1)P(\neg FP)}{P(H_1)P(\neg FP) + P(H_{2+})P(\neg FP)P(Q)}, \\ &= \frac{P(H_1)}{P(H_1) + P(H_{2+})P(Q)}. \end{aligned} \tag{15}$$

Note that $P(I|E_1)$ can be computed independently of $P(FP)$; this is because we are conditioning on E_1, which implies that no spurious lookups have occurred. Note,



(a) $P(I|E_1)$. (b) $P(I|E_2)$.

Fig. 5. AP3 attacks.

also, that as $P(Q)$ grows smaller, the fraction approaches closer to 1. As we noted in the Section 3.1, with the Castro et al's. secure lookup, $P(Q)$ is quite small, even for small f.

Figure 5(a) shows the attacker confidence as a function of the fraction of the nodes that are compromised for varying p, using $N = 1,000$, $b = 4$, $L = 16$. Our calculations show that to achieve $P(I|E_1) < 1/2$, we require that $f < 0.05$, which is much smaller than the previously computed limit of $f < 0.33$. Furthermore, the theoretical limit for the fraction of attackers that AP3 can tolerate can be computed by letting $p \to 1$, which is approximately 10% attackers. Again, this limit is much smaller than the conventional figure of 50%. This shows the fundamental tension that is encountered by AP3. The default Pastry mechanisms provide little defense against active adversaries who will try to disrupt the lookup process, dramatically increasing $P(H_1)$ and thus $P(I|H_{1+})$. Castro et al.'s suggested mechanisms solve this problem, but introduced another, as the lookup is no longer anonymous and can be observed by malicious nodes.

4.2. The E_i Attack

In addition to E_1, the adversary can use the observation that if there is a chain of lookups leading to the predecessor node, then the first node in the chain is more likely to be the initiator than any other node. For instance, we can define E_2 as the case, in which attackers observe a lookup by some node Q of the previous hop (P), but do not detect a lookup for Q. Furthermore, the previous hop (P) should not have looked up any other nodes. We now compute $P(I|E_2)$. Depending on the probabilities of $P(E_2 \wedge H_1)$ and $P(E_2 \wedge H_2)$, the attacker may guess that P or Q is the initiator of the path.

These probabilities will depend on the chance of a false-positive lookup detection, which in turn depends on the amount of lookup traffic elsewhere in the network. We define x to be the number of paths that are being constructed (by all nodes) at the same time as this one. A reasonable number for x is $N/100$, which means that during this path construction, 1% of all nodes also performed a concurrent path construction. A number much larger than this (e.g., $N/10$) would mean that nodes are spending a significant fraction of their time (10%) constructing paths, rather than using them for anonymous communication. Also, if any nodes in the network are not in active use, this will decrease x.

Given x, we can compute the false-positive probability α using the equation

$$\alpha = 1 - \left(\frac{N-1}{N}\right)^{x\left(1-(1-f)^{L+\log_{2^b} N}\right)}.$$

It is easy to see that as long as the false positive detection probability is small, $P(E_2 \wedge H_1) \ll P(E_2 \wedge H_2)$. Therefore, the attacker strategy here would be to guess the node (Q) looking up the previous hop to be the initiator. Therefore, $P(I|E_2 \wedge H_1) = 0$ and $P(I|E_2 \wedge H_{3+}) = 0$.

$$P(I|E_2) = \frac{P(I|E_2 \wedge H_2)P(E_2 \wedge H_2)}{P(E_2 \wedge H_1) + P(E_2 \wedge H_2) + P(E_2 \wedge H_{3+})}. \tag{16}$$

Figure 5(b) plots $P(I|E_2)$ as a function of f for varying p. The trend for $P(I|E_2)$ is very similar to our analysis of $P(I|E_1)$. Again, we can see that for $p = 0.75$, the maximum fraction of attackers that AP3 can handle while maintaining $P(I|E_2) < 1/2$ is only 5%. Due to lack of space, we have limited our analysis to only $P(I|E_1)$ and $P(I|E_2)$. In this sense, ours is a conservative analysis and the attackers can utilize many more observation points. For instance, one could define a general event E_i analogous to E_2. If the false positives are small, $P(I|E_i)$ can be approximated as

$$P(I|E_i) = \frac{P(H_i)}{P(H_i) + P(H_{(i+1)+})P(Q)}.$$

This formulation neglects false positives and is only an approximation. However, in practice, the approximation works quite well. In Figure 5(b), we can see that the results of the approximate model are quite close to the actual formulation that takes false positives into account.

Note that the metrics $P(I|E_1)$ and $P(I|E_2)$ are only indicative of the attacker confidence in identifying the initiator, given the observations E_1 and E_2. They do not consider the likelihood of the attackers observing E_1 and E_2. We use the entropy metric of anonymity [Diaz et al. 2002; Serjantov and Danezis 2002] to take this into account. The metric relies on computing the entropy of the distribution of possible initiators of a path. In the case of E_i, the probability that the identified node is the initiator is $P(I|E_i)$, and the probability assigned to any other node is $\frac{1-P(I|E_i)}{N-1}$.[2] Let $H(E_i)$ be the entropy of the system under the observation E_i. Then, the average entropy can be computed as

$$H = P(E_1)H(E_1) + P(E_2)H(E_2) + (1 - P(E_1) - P(E_2)) \log_2 N.$$

Figure 6 plots the entropy as a function of f, for varying p, using $N = 1,000$. Note that higher values of p have lower entropy, and can thus be considered to provide worse anonymity under the entropy metric. With higher path lengths, the observation E_2 (and $E_3, E_4, \ldots$) is more frequent, even though each observation has lower confidence. The latter effect dominates, highlighting one of the open questions in anonymity analysis: is it better to have an anonymity system that allows weak attacks frequently, or strong attacks rarely?

5. SALSA

We shall now analyze Salsa's path-building mechanism. For anonymous communication, a path is built between the initiator and the recipient via proxy routers (nodes). Layered encryption ensures that each node knows only its previous and next hop in the path. The nodes used for the paths are randomly selected from the global pool of nodes, even though each node has only local knowledge of a small subset of the network.

[2]This is a slight simplification; the entropy metric can take into account that, for example, in the case of E_2, P is more likely to be the initiator than a random node.

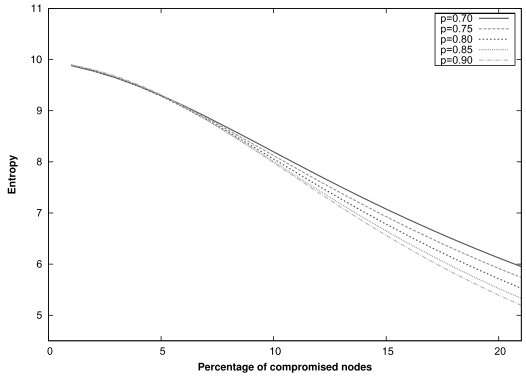
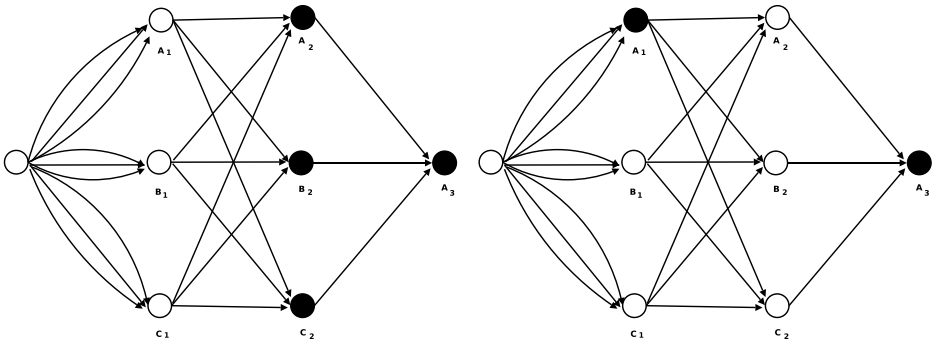


Fig. 6. Entropy as a function of f.



(a) Bridging an honest first stage. (b) Bridging an honest stage.

Fig. 7. Information leak attacks on Salsa.

5.1. Salsa Path-Building

To build a circuit, the initiator chooses r random IDs (Nambiar and Wright [2006], set $r = 3$) and redundantly looks up the corresponding nodes (called the first set/stage of nodes). Keys are established with each of these nodes. Each of the first set of nodes does a single lookup for r additional nodes (second set of nodes). A circuit is built to each of the nodes in the second group, relayed through one of the nodes in the first group. Again, the initiator instructs the second set of nodes (via the circuits) to do a lookup for a final node. One of the paths created between the first and the second set of nodes is selected, and the final node is added to the circuit. We use the parameter l to refer to the number of stages in the circuit (Nambiar and Wright [2006], set $l = 3$). Figure 7(a) depicts the Salsa path-building mechanism for $r = 3$ and $l = 3$. Note that redundant lookups are used only to look up the nodes in the first stage; later lookups rely on the redundancy in the path-building mechanism itself.

5.2. Active Path Compromise Attacks on Salsa

Active attacks on the lookup mechanism can bias the probability that nodes involved in Salsa's path-building mechanism are compromised. Borisov et al. [2007] noted that Salsa path-building is also subject to a public key modification attack.[3] If all the nodes

[3]Their analysis did not take into account the lookup bias.

in a particular stage are compromised, they can modify the public keys of the next set of nodes being looked up. This attack defeats Salsa's bounds check algorithm that ensures the IP address is within the right range, since it cannot detect an incorrect public key. Also, since the traffic toward the node whose public key has been modified is forwarded via corrupt nodes, the attackers are guaranteed to intercept the messages. They can then complete the path-building process by emulating all remaining stages (and hence, the last node). The public key modification attack and attacks on Salsa lookup mechanism are active attacks. By end-to-end timing analysis, the path will be compromised if the first and last nodes in the circuit are compromised. Conventional analysis of anonymous communication typically focuses on minimizing the chance of path compromise attacks. By increasing the redundancy in the path-building mechanism, this chance can be minimized, as increasing r decreases the chance of both active attacks on lookups, as well as public key modification attacks.

We now describe three types of passive information leak attacks on Salsa. We also show that increasing redundancy increases the effectiveness of the information leak attacks, resulting in a tradeoff between robustness against active attacks and passive information leak attacks.

5.3. Conventional Continuous Stage Attack

A path in Salsa can be compromised if there is at least one attacker node in every stage of the path. Suppose that there are attacker nodes A_1, A_2, A_3 in the three stages, respectively. In the path-building mechanism, a node performs a lookup for all r nodes in the following stage implying that A_1 would have looked up A_2, and A_2 would have looked up A_3. Hence the attacker can easily (passively) bridge the first and last stages, thereby compromising the anonymity of the system. (This attack was mentioned by Nambiar and Wright [2006]). Note that if we increase redundancy as per conventional analysis, the effectiveness of the continuous stage attack also increases. This is because increasing redundancy increases the chance that attackers are present in each stage (which is $1-(1-f)^r$), giving them more opportunities to launch this attack. Next, we describe two new bridging attacks also based on information leaks from lookups.

5.4. Bridging an Honest First Stage

This attack is based on the observation that an initiator performs redundant lookups for the nodes in the first stage. If the adversary can deduce the identities of the nodes in the first stage (they need not be compromised) and can detect any of the initiator's redundant lookups for nodes in the first stage, the anonymity of the system is compromised. Consider the Figure 7(a); malicious nodes, are depicted in black. The first stage (A_1, B_1, C_1) is comprised solely of honest nodes, the second stage (A_2, B_2, C_2) has all malicious nodes; and the third stage, node A_3, is also compromised. The attackers know the identities of $A_1, B_1,$ and C_1 because of key establishment with them. If they detect a node performing a lookup for either A_1, B_1, or C_1, they can identify that node as the initiator. Since the initiator performs nine lookups for the first-stage nodes, the probability of detecting this initiator is $1-(1-f)^9$, which translates into a probability of 0.87 for $f = 0.2$. A similar attack strategy is applicable when only two, or even one, node in the second stage is compromised. In the latter scenario, the second stage knows the identity of only a single node in the first stage, and if the initiator is detected looking up that node, then the path is compromised. This occurs with probability $1-(1-f)^3$, which is 0.49 for $f = 0.2$. Similar to the continuous stage attack, notice that an increase in r increases the probability that attackers can detect a lookup by the initiator for the first node.

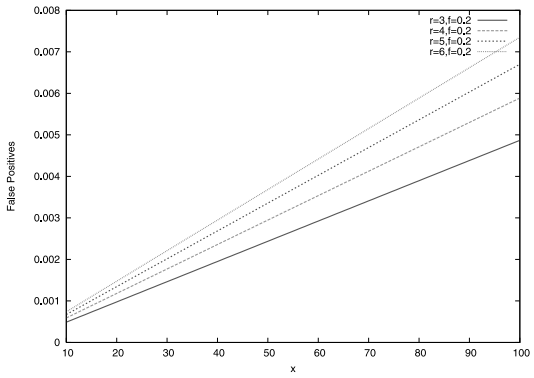


Fig. 8. False positives in bridging an honest first stage.

It is important to note that there are some false positives in the attack. The false positives occur when a node (say A_1) in the first stage is involved in building more than one path. In such a scenario, more than one node will lookup A_1, and the attackers may detect a lookup for A_1 not done by the actual initiator. Using the variable x to model the amount of lookup traffic by other nodes, as in Section 4.2, we can compute the false positive probability as

$$1 - \left(\frac{N-1}{N}\right)^{x\left(1-(1-f)^r\right)}.$$

Figure 8 depicts the false-positive probability for varying r, using $f = 0.2$, $N = 1{,}000$. Note that for $x < \frac{N}{100}$, the false positive probability is less than 0.1%.

5.5. Bridging an Honest Stage

Salsa is also vulnerable to a bridging attack in which attacker nodes separated by a stage with all honest nodes are able to deduce that they are on the same path. Consider the arrangement of nodes depicted in Figure 7(b). The first stage has one malicious node A_1; the second stage consists solely of honest nodes; and the last node A_3 is compromised. A_1 knows the identities of all three nodes in the second stage, as it has performed a lookup for them. Also, as part of the path-building mechanism, one of the nodes in the second stage will establish a key with the compromised third-stage node, A_3. In such a scenario, A_1 and A_3 can deduce that they are part of the same path, as they both observe a common honest node. Similarly, if any of the nodes in the first stage are compromised and the last node is compromised, the path is compromised. In such an attack, the compromised nodes in the first stage need not be selected as relays. Again, recall that increasing r increases the chance of an attacker being present in a stage, resulting in a higher probability of bridging an honest stage. The probability of false positives in this scenario can be analyzed as $1 - \left(\frac{N-1}{N}\right)^x$, which for $x = N/100$ and $N = 1{,}000$ is less than 1%.

5.6. Results

We now present experimental results for active path compromise attacks and information leak attacks on Salsa. Our results have been computed by modeling the Salsa path-building mechanism as a stochastic activity network in the Möbius framework [Daly et al. 2000]. For a fixed f and r, the input to the model is the lookup bias,

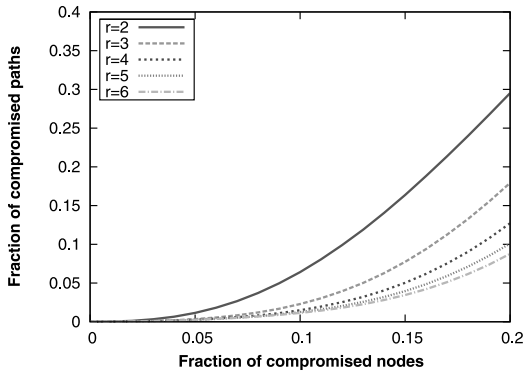


Fig. 9. Conventional path compromise attacks: increasing redundancy counters active attacks.

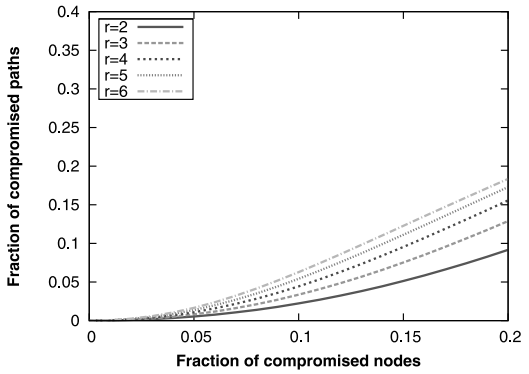


Fig. 10. Information leak attacks: increasing redundancy makes the passive adversary stronger.

which was computed using the Salsa simulator [Nambiar and Wright 2007], with simulation parameters $N = 1{,}000$, $|G| = 128$.

Figure 9 shows the chance of active path compromise attacks on Salsa for varying levels of redundancy. It is easy to see that increasing r reduces the fraction of compromised paths. For instance, at $f = 0.2$, 17% paths are compromised using $r = 3$. The corresponding value for $r = 6$ is approximately 8%. This is not surprising, as increasing r reduces the chance of both active attacks on lookups and attacks involving public-key modification.

The continuous stage attack and both our bridging attacks are examples of passive attacks. Figure 10 shows the fraction of compromised paths under the passive attacks. We can see that an increase in r increases the effectiveness of the passive attacks and is detrimental to anonymity. For 20% of attackers, even for a small value of $r = 3$, the initiator can be identified with probability 0.125. Higher values of r can increase the probability of identifying the initiator to over 0.15. Note also that the bridging attack significantly improves upon the previous attacks on Salsa: using only the continuous stage attack for $r = 3$, $f = 0.2$, anonymity is broken with a probability of only 0.048—less than half of what is possible with bridging.

The active path compromise attacks can be combined with passive information leak attacks. Figure 11 shows the fraction of compromised paths for all passive and active attacks. An interesting trend is observed in which increasing redundancy (beyond $r = 2$) is detrimental to security for small values of f. This is in sharp contrast to

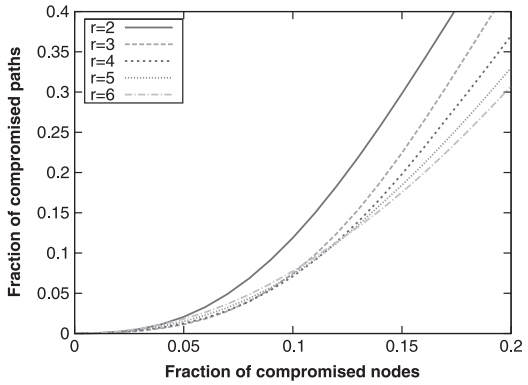


Fig. 11. All conventional and information leak attacks: for maximal anonymity, $r = 3$ is optimal for small f. Note that there is a crossover point at $f = 0.1$, when $r = 6$ becomes optimal.

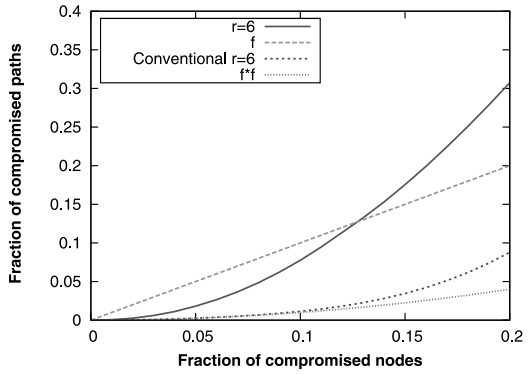


Fig. 12. Comparison of all attacks with conventional active attacks: note that for $f > 0.12$, fraction of compromised paths is greater than f.

conventional analysis; the inclusion of information leak attacks have made the effect of passive attacks more dominant over the effect of active attacks. There is a crossover point at about 10% malicious nodes, after which increasing r reduces the probability of path compromise. This is because active attacks are dominant for higher values of f. Note that $r = 2$ results in significantly worse security because of poor resilience to both lookup attacks and public key modification attacks.

This shows the tension between passive and active attacks. There is an inherent redundancy in Salsa path-building mechanisms to counter active attacks. However, the redundancy makes the passive adversary stronger and provides more opportunities for attack. From Figure 12 we can see that by conventional analysis, security provided by Salsa is close to that of Tor (f^2). With our information leak attacks taken into account, for $f > 0.12$, the security provided by Salsa is even worse than f.

5.7. Improvements to Salsa

We next consider whether simple changes to Salsa's mechanisms would provide a defense against our attacks. First, we consider Salsa using a PKI, as in AP3. The public key modification attack would no longer work; however, other active attacks on the lookup mechanism and our passive information leak attacks would still apply. Figure 13 depicts the probability of identifying the initiator under all active and

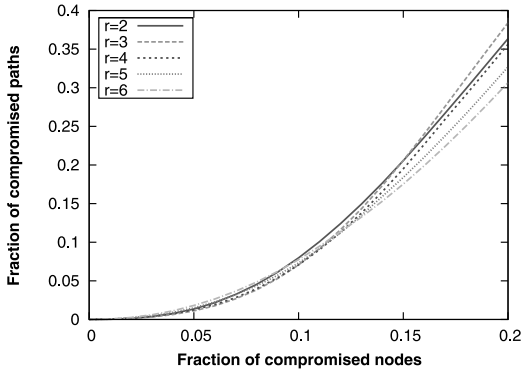


Fig. 13. Salsa with a PKI—All conventional and information leak attacks. Even with a PKI, the security of Salsa is much worse as compared to conventional analysis.

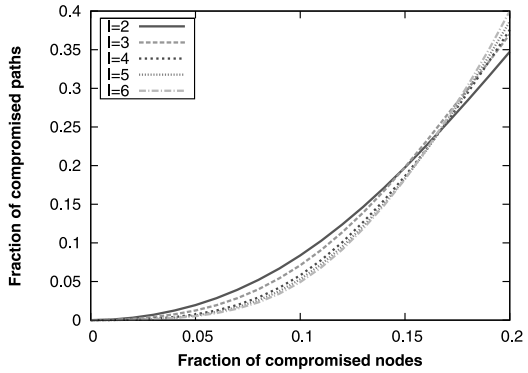


Fig. 14. Effect of varying the path length: note that there is only limited benefit of increasing path length.

passive attacks in Salsa with PKI. Again, we can see the tension between active and passive attacks. With the public key modification attack gone, $r = 2$ becomes a more reasonable parameter, but even with a PKI, the fraction of compromised paths increases from 8% under conventional active attacks to more than 30% with our information leak attacks taken into account.

Finally, we explore the effect of increasing the path length (l) on the anonymity of Salsa. Figure 14 depicts the probability of identifying the initiator for varying values of l. There is an interesting tradeoff in increasing the path length. On one hand, increasing l reduces the chance of information leak attacks, because the attacker needs to bridge all stages. On the other hand, increasing l gives attackers more opportunities to launch active attacks, thereby increasing the probability that the last node is compromised, which in turn gives attackers more observation points. This is basically a cascading effect: the presence of a malicious node in each stage increases the probability of the presence of malicious nodes in the next stage. For small values of f, passive attacks are stronger, therefore increasing l increases security, but for higher f, the active attacks and the cascading affect are dominant, therefore increasing l decreases security.

We have proposed passive bridging attacks on Salsa that are based on information leaks from lookups, and can be launched by a partial adversary. Moreover, we have shown a tradeoff between defenses against active and passive attacks. Even at the

optimal point in the tradeoff, the anonymity provided by the system is significantly worse than what was previously thought. This tradeoff is present even in Salsa with a PKI. Moreover, increasing path length in Salsa has only a limited benefit on user anonymity.

6. AN ENTROPY-BASED APPROACH FOR INFORMATION LEAKS

So far we had considered lookup anonymity in Salsa to be compromised only if the first hop (local contact) is malicious. However, information leaks also exist when any of the nodes in the lookup path are malicious, not just the first hop. The difference is that when the first hop is malicious, the lookup initiator is precisely identified, whereas in other cases, the attacker only learns some probabilistic information. We now present an analysis of this information leak, where instead of using a binary metric of identifying the lookup initiator, we use an entropy-based anonymity metric. This metric considers the distribution of potential initiators of the lookup (as computed by the attackers) and computes its Shannon entropy as

$$H_{Shannon}(I) = -\sum_i p_i \log_2 p_i, \tag{17}$$

where p_i is the probability that node i was the initiator of the lookup. Under some observation o, we can compute the probability distribution, given o, and compute the corresponding entropy $H(I|o)$. To model the entropy of the lookup as a whole, we compute a weighted average of the entropy for each observation (including the null observation), such that

$$H(I|O) = \sum_{o \in O} P(o)H(I|o), \tag{18}$$

where $P(o)$ is the probability of observation o occurring, and O is the set of all observations. This is also known as the conditional entropy of I based on observing O.

6.1. Single Lookup

When the lookup is not intercepted by the adversary (null observation), the attacker clearly does not learn any information and the entropy is $\log(1-f)N$. Now, let us consider the case when the lookup is intercepted by the adversary. The adversary can approximate the identity of the initiator by using the observation o that the previous hop p in the lookup path is y *levels* away from it in the binary tree structure. Thus we have

$$H(I|O) = \sum_{y,p} P(O = \langle y, p \rangle)H(I|O = \langle y, p \rangle). \tag{19}$$

To compute the entropy of the lookup, we need to compute $P(O = \langle y, p \rangle)$ and $H(I|O = \langle y, p \rangle)$. Let us first focus on $P(O = \langle y, p \rangle)$. We can decompose $P(O = \langle y, p \rangle)$ by conditioning on the the event $I = i$. We have

$$P(O = \langle y, p \rangle) = \sum_{i \text{ honest}} P(O = \langle y, p \rangle | I = i) \cdot P(I = i), \tag{20}$$

where $P(I = i)$ is the prior probability of node I being the initiator, given by

$$P(I = i) = \frac{1}{(1-f)N}. \tag{21}$$

Note that in this analysis, we have conservatively assumed that all users have no a priori linkability to their traffic. We now compute $P(O = \langle y, p \rangle | I = i)$. Let us denote

the distance between node i and the target, in terms of binary tree levels, as $D = d_i$. In the case when $y = 0$, $P(O = \langle y, p\rangle | I = i)$ is simply equal to the probability of the first hop being malicious (f) when $p = i$.

Next, we have the observation that a jump of size y relative to the malicious hop has a previous hop which is y levels away from the target node. This means that when $d_i = y$, then $P(O = \langle y, p\rangle | I = i)$ is equivalent to a jump from the initiator's group being intercepted by a malicious node. The probability of a particular node p being selected as the first hop in the initiator's group is $\frac{|G|}{N\cdot(1-f)-|G|}$ (considering only honest nodes and excluding the initiator). The probability of the jump being intercepted at the second hop is f, and the probability of observing y under these constraints is $\frac{2^{y-1}}{|G|}$. To sum up, this event happens with probability $\frac{|G|}{N\cdot(1-f)-|G|} \cdot f \cdot \frac{2^{y-1}}{|G|}$ when p is in the initiators group, and with probability 0 otherwise.

Lastly, let us consider the case $y < d_i$. If we suppose that the lookup has traversed l nodes so far (not including the final malicious hop), then we require that these l nodes be honest, and the final node is malicious. This occurs with probability $(1-f)^l \cdot f$. We know that the first hop is always in the initiator's group, and to get a jump of y, the lookup also traverses the subtree which is y levels away from the target (the selection probability of which is $\frac{1}{2}$). Furthermore, the probability of selecting a particular node p in this subtree is $\frac{1}{2^{y-1}} \cdot \frac{|G|}{N(1-f)}$. With these constraints, the probability of the lookup traversing the remaining $l-2$ hops can be computed as a selection problem of choosing $l-2$ subtrees out of the possible $d-y-1$, where the probability of selection is $\frac{1}{2}$. This is a binomial distribution with probability $\binom{d-y-1}{l-2} \cdot \left(\frac{1}{2}\right)^{d-y-1}$. Combining all this, we have

$$P(O = \langle y, p\rangle | I = i) = \begin{cases} f & y = 0, i = p \\ \frac{|G|}{N\cdot(1-f)-|G|} \cdot f \cdot \frac{2^{y-1}}{|G|} & i, p \in \text{same group} \\ \sum_{l=2}^{d-y+1} (1-f)^l \cdot f \cdot \frac{1}{2} \cdot \frac{1}{2^{y-1}} \cdot \frac{|G|}{N(1-f)}. & \\ \quad \binom{d-y-1}{l-2} \cdot (\frac{1}{2})^{d-y-1} \cdot \frac{2^{y-1}}{|G|} & \text{otherwise.} \end{cases} \tag{22}$$

Using $P(I = i)$ and $P(O = \langle y, p\rangle | I = i)$ from Equations (21) and (22), we can now compute $P(O = \langle y, p\rangle)$ from Equation (20).

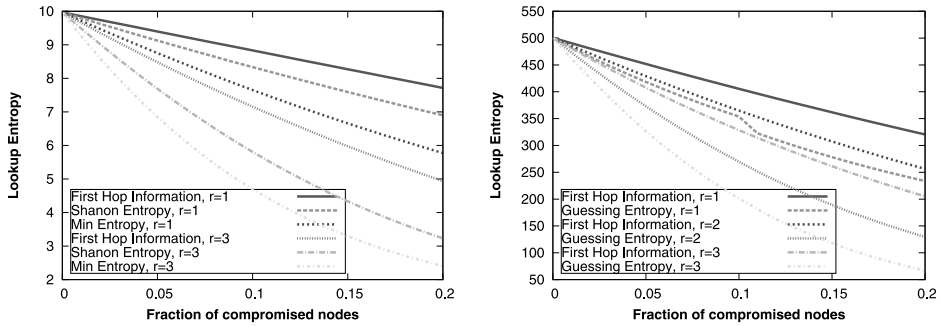
Let us now compute $H(I|O = \langle y, p\rangle)$. By definition, we have

$$H(I|O = \langle y, p\rangle) = -\sum_{i \text{ honest}} P(I = i|O = \langle y, p\rangle) \log P(I = i|O = \langle y, p\rangle). \tag{23}$$

Since we have already computed $P(O = \langle y, p\rangle | I = i)$, $P(I = i)$, and $P(O = \langle y, p\rangle)$ in Equations (22), (21), and (20), respectively, we can use Bayesian inference to compute $P(I = i|O = \langle y, p\rangle)$ as

$$P(I = i|O = \langle y, p\rangle) = \frac{P(O = \langle y, p\rangle | I = i) \cdot P(I = i)}{P(O = \langle y, p\rangle)}. \tag{24}$$

By using $P(O = \langle y, p\rangle)$ from Equation (20) and $H(I|O = \langle y, p\rangle)$ from Equation (23), we can compute the entropy of the lookup from Equation (19).



(a) Shannon entropy and min-entropy. (b) Guessing entropy.

Fig. 15. Lookup entropy.

6.2. Redundant Lookups

Let us denote the attackers' observations for the r redundant lookups as $o_1 = \langle y_1, p_1\rangle, \ldots, o_r = \langle y_r, p_r\rangle$.

$$H(I|O) = \sum_{y_1,p_1} \cdots \sum_{y_r,p_r} P(\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r}) H(I|\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r}) \tag{25}$$

Similar to the case of single lookup, we can condition the probability $P(\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r})$ on the event $I = i$. Using the observation that the redundant lookups are independent, given $I = i$, we can compute $P(\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r})$ as

$$P(\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r}) = \sum_{i \text{ honest}} P(\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r} | I = i) \cdot P(I = i), \tag{26a}$$

$$P(\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r}) = \sum_{i \text{ honest}} \prod_{k=1}^{r} P(o_k = \langle y_k, p_k\rangle | I = i) \cdot P(I = i), \tag{26b}$$

where $P(O = \langle y, p\rangle | I = i)$ and $P(I = i)$ are given by Equations (22) and (21). Let us now compute $H(I|\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r})$.

$$H(I|\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r}) = - \sum_{i \text{ honest}} P(I = i|\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r}) \log P(I = i|\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r}). \tag{27}$$

Again, we make use of Bayesian inference to combine information from multiple observations as follows.

$$P(I = i|\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r}) = \frac{P(\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r} | I = i) \cdot P(I = i)}{P(\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r})}, \tag{28a}$$

$$= \frac{\prod_{k=1}^{r} P(o_k = \langle y_k, p_k\rangle | I = i) \cdot P(I = i)}{P(\{o_j = \langle y_j, p_j\rangle\}_{j=1}^{r})}. \tag{28b}$$

Finally, we can use Equation (25) to compute the entropy of redundant lookups.

Figure 15(a) plots the entropy of the lookup as a function of the fraction of compromised nodes in the system, for varying values of redundancy. The input parameters for our model were $N = 1{,}000$, $|G| = 128$. We can see that by considering the all possible information leaks from the lookup, the lookup entropy is considerably reduced, as compared to the scenario where we considered information leaks only from the first

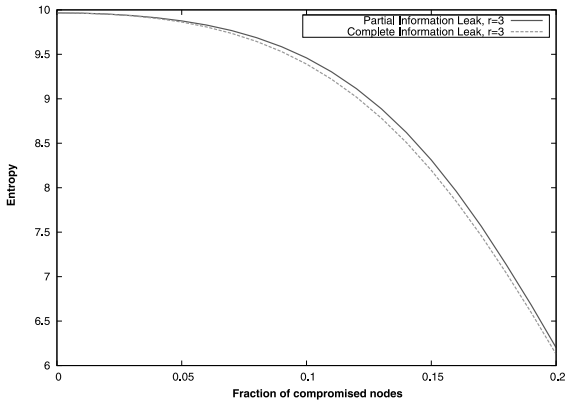


Fig. 16. Path entropy.

hop. When the fraction of compromised nodes is 20% and the redundancy level is $r = 3$, then using the complete information reduces the lookup entropy from about 5 bits to 3.2 bits (Shannon entropy). In addition to Shannon entropy, Figure 15(a) also presents results for min-entropy. The min-entropy is computed as

$$H_{Min}(I) = -\log_2 \max p_i. \tag{29}$$

We can see that for $f = 0.2, r = 3$, using complete information reduces the min-entropy by more than half from 5 bits to 2.4 bits. Finally, we also present the guessing entropy for the Salsa lookup in Figure 15(b). The guessing entropy can be computed by first arranging the nodes in decreasing order of probability p_i and then using the equation

$$H_{Guessing}(I) = \Sigma_i p_i \cdot i. \tag{30}$$

We can see that $f = 0.2, r = 3$, using complete information reduces the guessing entropy by more than a third from 210 guesses to only 66 guesses. Our analysis illustrates that our security evaluation for Salsa's path-building mechanism is a conservative analysis, and the actual anonymity loss due to information leaks via lookups would be even greater than our results suggest.

6.3. Path Construction

Our entropy-based analysis of lookups suggests that the anonymity provided by the path-construction mechanism is likely to be even lower than our results shown in Section 5. This is because our earlier results on path construction considered only scenarios where exact identification of the initiator is possible and ignored the significant amount of probabilistic information that an adversary has.

Consider our attack that involves bridging an honest first stage. In this setting, the adversary controls the final node and has knowledge of at least one node in the first stage. In our earlier results, we had considered the user anonymity to be compromised if the adversary is able to exactly identify the initiators based on its lookups for the node(s) in the first stage. Instead, we can now compute the initiator entropy based on its lookups for the first-stage nodes. If the adversary knows $x < r$ nodes in the first stage (and the last node is compromised), then the initiator entropy is equivalent to the lookup entropy with redundancy parameter $x \cdot r$.

Figure 16 shows the reduction in the anonymity (Shannon entropy) based on the additional probabilistic information, while bridging the first honest stage alone. We

have left a complete analysis of Salsa's path-building mechanism using the entropy based metric as part of future work.

7. RELATED WORK

Secure routing in peer-to-peer networks has been the subject of much research [Castro et al. 2002; Kapadia and Triandopoulos 2008; Nambiar and Wright 2006; Sit and Morris 2002; Wallach 2002]. We studied lookup mechanisms proposed by Castro et al. [2002] and Nambiar and Wright [2006], focusing on the information leak from lookups, and observed a tradeoff between security and anonymity of a lookup. Kapadia and Triandopoulos recently proposed Halo [2008], which is also based on redundant routing and exhibits a similar tradeoff. Moreover, it uses very high redundancy levels, as compared to Salsa, and would make our information leak attacks more effective. There have been some attempts to add anonymity to a lookup. Borisov [2005] proposed an anonymous DHT based on Koorde [Kaashoek and Karger 2003], which performs a randomized routing phase before an actual lookup. Ciaccio [2006] proposed the use of imprecise routing in DHTs to improve sender anonymity. These lookups were designed to be anonymous but not secure: an active adversary could easily subvert the path of the lookup. As such, neither lookup mechanism can be used to build anonymous circuits. Recently, Panchenko et al. [2009] proposed to build anonymity into a secure lookup mechanism, but Wang et al. [2010] showed that it is possible to compromise lookup anonymity.

Danezis and Clayton [2006] studied attacks on peer discovery and route setup in anonymous peer-to-peer networks. They show that if the attacker learns the subset of nodes known to the initiator (by observing lookups, for example), its routes can be fingerprinted, unless the initiator knows about the vast majority of the network. Danezis and Syverson [2007] extend this work to observe that an attacker who learns that certain nodes are unknown to the initiator can carry out attacks, as well, and separate traffic going through a relay node. These attacks are similar in spirit to the ones we propose, but rather than absolute knowledge of the initiator's routing state, we use probabilistic inferences based on observed lookups. Recently, Bauer et al. [2007] proposed a bridging attack in Tor where attacker nodes sandwiching an honest node can correlate the path, even before a packet is sent. This attack is similar to our bridging attack on Salsa, except that we also utilize information leaks from lookups and consider the issue of false positives.

Reiter and Rubin [1998] proposed the predecessor attack, which was later extended by Wright et al. [2002, 2003, 2004]. In this attack, an attacker tracks an identifiable stream of communication over multiple communication rounds and logs the preceding node on the path. To identify the initiator, the attacker uses the observation that the initiator is more likely to be the predecessor than any other node in the network. For peer-to-peer anonymous communication systems like Salsa, the number of rounds required by predecessor attacks to identify the initiator with high probability is inversely proportional to the probability of success of end-to-end timing analysis. This means that defenses that minimize the chance of both first and last nodes being attackers also increase resilience against predecessor attacks. In this article, we only analyzed the scenario in which an initiator constructs a single communication path to the destination. We leave a complete analysis for multiple communication rounds as part of future work.

Similar to predecessor attacks, there is a thread of research that deals with degradation of anonymity over a period of time. Berthold et al. [2000] and Raymond [2000] propose intersection attacks that aim to compromise sender anonymity by intersecting

sets of user's that were active at the time the intercepted message was sent, over multiple communication rounds. Similarly, Kesdogan et al. [2002] use intersection to find recipients of a given users message. A statistical version of this attack was proposed by Danezis [2003] and later extended by Mathewson and Dingledine [2004]. Information leaks in P2P systems can allow even a partial adversary to make observations about a large fraction of lookups and path-building, and can, therefore, form a basis of effective statistical intersection and disclosure attacks.

An important point of our article is that, when building anonymous systems, it is important not to abstract away the properties of the system that can affect anonymity. Our analysis of AP3 is an example of how composition of two designs that are secure individually [Castro et al. 2002; Reiter and Rubin 1998] creates new vulnerabilities. Similar in spirit to ours, a lot of recent research has focused on details abstracted away by conventional analysis models to break the anonymity of the system. Such details include congestion and interference [Back et al. 2001; Murdoch and Danezis 2005], clock skew [Murdoch 2006], heterogeneous path latency [Back et al. 2001; Hopper et al. 2007], the ability to monitor Internet exchanges [Murdoch and Zieliński 2007], and reliability [Borisov et al. 2007].

8. CONCLUSION

We have analyzed information leaks in the lookup mechanisms of peer-to-peer anonymous communications systems. Existing defenses against active attacks typically use redundant messages, which enable a relatively small fraction of attackers to observe a large number of lookups in the network. Attackers are thus able to act as a near-global passive adversary and use this to break the anonymity of the system.

We have shown how attacks based on information leaks from lookups can be used to break the probable innocence guarantees in AP3. We computed the limit on the number of attackers that AP3 can handle while providing probable innocence as only 5% in the typical case, while the theoretical limit with increased path lengths is 10%. This is in contrast to the conventional analysis, which puts these figures at 33% and 50%, respectively. A small fraction of malicious nodes can therefore compromise the security of AP3. An important lesson learned from the AP3 analysis is that the composition of a secure DHT lookup mechanism with an anonymous communication protocol (as has been considered in other work [Sherr et al. 2007]) should be carefully analyzed, as it is likely to introduce additional vulnerabilities.

We have also analyzed the security of Salsa under both active and passive attacks. We have demonstrated the tension that exists between defending against both active and passive adversaries. Defending against active adversaries requires higher redundancy, which increases the threat of passive attacks. Salsa was previously reported to tolerate up to 20% compromised nodes, but our results show that, with information leaks taken into account, over a quarter of all tunnels are compromised. Moreover, we show that the tension between active and passive attacks exists even if Salsa were to use a PKI. Also, increasing path lengths to counter our passive attacks only has a limited benefit, and in some cases, it even reduces anonymity.

Our results demonstrate that information leaks are an important part of anonymity analysis of a system.

ACKNOWLEDGMENTS

We would like to thank Arjun Nambiar, Nayantara Malesh, and Matthew Wright for providing us with the Salsa simulator and for helpful discussions about the Salsa algorithms. We are also grateful to George Danezis and Matthew Wright for their comments on earlier versions of this paper.

REFERENCES

BACK, A., MÖLLER, U., AND STIGLIC, A. 2001. Traffic analysis attacks and trade-offs in anonymity providing systems. In *Proceedings of the Information Hiding Workshop*. I. S. Moskowitz Ed., Lecture Notes in Computer Science, vol. 2137. Springer, 245–247.

BAUER, K., MCCOY, D., GRUNWALD, D., KOHNO, T., AND SICKER, D. 2007. Low-resource routing attacks against Tor. In *Proceedings of the ACM Workshop on Privacy in the Electronic Society*. T. Yu Ed., ACM, New York, NY, 11–20.

BELLOVIN, S. M. AND WAGNER, D. A., Eds. 2003. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, Los Alamitos, CA.

BERTHOLD, O., FEDERRATH, H., AND KÖHNTOPP, M. 2000. Project "anonymity and unobservability in the Internet". In *Proceedings of the 10th Conference on Computers, Freedom and Privacy*. L. Cranor Ed., ACM, New York, NY, 57–65.

BORISOV, N. 2005. Anonymous routing in structured peer-to-peer overlays. Ph.D. thesis, UC Berkeley.

BORISOV, N., DANEZIS, G., MITTAL, P., AND TABRIZ, P. 2007. Denial of service or denial of security? How attacks on reliability can compromise anonymity. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*. 92–102.

BOUCHER, P., SHOSTACK, A., AND GOLDBERG, I. 2000. Freedom systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc.

CASTRO, M., DRUSCHEL, P., GANESH, A., ROWSTRON, A., AND WALLACH, D. S. 2002. Secure routing for structured peer-to-peer overlay networks. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*. D. Culler and P. Druschel Eds., USENIX, Berkeley, CA, 299–314.

CIACCIO, G. 2006. Improving sender anonymity in a structured overlay with imprecise routing. In *Proceedings of the 6th Workshop on Privacy Enhancing Technologies*. 190–207.

CLARKE, I., SANDBERG, O., WILEY, B., AND HONG, T. W. 2001. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of the International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*. Springer Verlag, Berlin, 46–66.

COOKE, E., JAHANIAN, F., AND MCPHERSON, D. 2005. The zombie roundup: Understanding, detecting, and disrupting botnets. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop*. USENIX Association, Berkeley, CA, 6–6.

DALY, D., DEAVOURS, D. D., DOYLE, J. M., WEBSTER, P. G., AND SANDERS, W. H. 2000. Möbius: An extensible tool for performance and dependability modeling. In *Computer Performance Evaluation. Modelling Techniques and Tools*. B. R. Haverkort, H. C. Bohnenkamp, and C. U. Smith Eds., Lecture Notes in Computer Science, vol. 1786. Springer, 332–336.

DANEZIS, G. 2003. Statistical disclosure attacks: Traffic confirmation in open environments. In *Proceedings of the IFIP TC11 18th International Conference on Information Security (SEC)*. D. Gritzalis, S. di Vimercati, P. Samarati, and S. Katsikas Eds., 421–426.

DANEZIS, G. AND CLAYTON, R. 2006. Route fingerprinting in anonymous communications. In *Proceedings of the IEEE Conference on Peer-to-Peer Computing*. IEEE Computer Society, Los Alamitos, CA, 69–72.

DANEZIS, G. AND GOLLE, P., Eds. 2006. In *Proceedings of the Privacy Enhancing Technologies*. Lecture Notes in Computer Science, vol. 4258. Springer, Berlin.

DANEZIS, G. AND SYVERSON, P. 2007. Bridging and fingerprinting: Epistemic attacks on route selection. In *Proceedings of the Privacy Enhancing Technologies Symposium*. N. Borisov and I. Goldberg Eds., Lecture Notes in Computer Science, vol. 5134. Springer, Berlin, 151–166.

DANEZIS, G., DINGLEDINE, R., AND MATHEWSON, N. 2003. Mixminion: Design of a Type III anonymous remailer protocol. In *Proceedings of the IEEE Symposium on Security and Privacy*. 2–15.

DIAZ, C., SEYS, S., CLAESSENS, J., AND PRENEEL, B. 2002. Towards measuring anonymity. In *Proceedings of the Workshop on Privacy Enhancing Technologies*. 184–188.

DINGLEDINE, R. AND SYVERSON, P., Eds. 2002. In *Proceedings of the Workshop on Privacy Enhancing Technologies*. Lecture Notes in Computer Science, vol. 2482. Springer.

DINGLEDINE, R., MATHEWSON, N., AND SYVERSON, P. 2004. Tor: The second-generation onion router. In *Proceedings of the USENIX Security Symposium*. M. Blaze Ed., USENIX Association, Berkeley, CA, 303–320.

DOUCEUR, J. 2002. The sybil attack. In *Proceedings of the 1st Workshop on Peer-to-Peer Systems*. 251–260.

DRUSCHEL, P., KAASHOEK, F., AND ROWSTRON, A., Eds. 2002. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*. Lecture Notes in Computer Science, vol. 2429. Springer, Berlin.

FEDERRATH, H., Ed. 2000. In *Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability*. Lecture Notes in Computer Science, vol. 2009. Springer, Berlin.

FREEDMAN, M. J. AND MORRIS, R. 2002. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the ACM Conference on Computer and Communications Security*. R. Sandhu Ed., ACM, New York, NY, 193–206.

GOODIN, D. 2007. Tor at heart of embassy passwords leak. *The Register*.

HOLZ, T., STEINER, M., DAHL, F., BIERSACK, E., AND FREILING, F. 2008. Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm. In *Proceedings of the 1st USENIX Workshop on Large-scale Exploits and Emergent Threats*. F. Monrose Ed., USENIX Association, Berkeley, CA.

HOPPER, N., VASSERMAN, E. Y., AND CHAN-TIN, E. 2007. How much anonymity does network latency leak? In *Proceedings of the 14th ACM Conference on Computer and Communications Security*. 82–91.

I2P. 2003. I2P anonymous network. `http://www.i2p2.de/index.html`.

KAASHOEK, M. F. AND KARGER, D. R. 2003. Koorde: A simple degree-optimal distributed hash table. In *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS)*. F. Kaashoek and I. Stoica Eds., Lecture Notes in Computer Science, vol. 2735. Springer, Berlin, 98–107.

KAPADIA, A. AND TRIANDOPOULOS, N. 2008. Halo: High-assurance locate for distributed hash tables. In *Proceedings of the Network and Distributed System Security Symposium*. C. Cowan and G. Vigna Eds., Internet Society, Reston, VA, 61–79.

KESDOGAN, D., AGRAWAL, D., AND PENZ, S. 2002. Limits of anonymity in open environments. In *Proceedings of the Information Hiding Workshop*. F. A. Petitcolas Ed., Lecture Notes in Computer Science, vol. 2578. Springer, Berlin, 53–69.

MATHEWSON, N. AND DINGLEDINE, R. 2004. Practical traffic analysis: Extending and resisting statistical disclosure. In *Proceedings of the Workshop on Privacy Enhancing Technologies*. D. Martin and A. Serjantov Eds., Lecture Notes in Computer Science, vol. 3424. Springer, Berlin, 17–24.

MCLACHLAN, J., TRAN, A., HOPPER, N., AND KIM, Y. 2009. Scalable onion routing with torsk. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09)*. ACM, New York, NY, 590–599.

MISLOVE, A., OBEROI, G., POST, A., REIS, C., DRUSCHEL, P., AND WALLACH, D. S. 2004. AP3: Cooperative, decentralized anonymous communication. In *Proceedings of the ACM SIGOPS European Workshop*. M. Castro Ed., ACM, New York, NY, 30.

MITTAL, P. AND BORISOV, N. 2009. Shadowwalker: Peer-to-peer anonymous communication using redundant structured topologies. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09)*. ACM, New York, NY, 161–172.

MÖLLER, U., COTTRELL, L., PALFRADER, P., AND SASSAMAN, L. 2003. Mixmaster Protocol—version 2. IETF Internet Draft.

MURDOCH, S. J. 2006. Hot or not: Revealing hidden services by their clock skew. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*. 27–36.

MURDOCH, S. J. AND DANEZIS, G. 2005. Low-cost traffic analysis of Tor. In *Proceedings of the IEEE Symposium on Security and Privacy*. V. Paxson and M. Waidner Eds., IEEE Computer Society Press, Los Alamitos, CA, 183–195.

MURDOCH, S. J. AND ZIELIŃSKI, P. 2007. Sampled traffic analysis by Internet-exchange-level adversaries. In *Proceedings of the Privacy Enhancing Technologies Symposium*. N. Borisov and P. Golle Eds., Lecture Notes in Computer Science, vol. 4776. Springer, 167–183.

NAMBIAR, A. AND WRIGHT, M. 2006. Salsa: A structured approach to large-scale anonymity. In *Proceedings of the 13th ACM Conference on Computer and Communications Secuity*. 17–26.

NAMBIAR, A. AND WRIGHT, M. 2007. The Salsa simulator. `http://ranger.uta.edu/~mwright/code/salsa-sims.zip`.

PANCHENKO, A., RICHTER, S., AND RACHE, A. 2009. Nisan: Network information service for anonymization networks. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09)*. ACM, New York, NY, 141–150.

RAJAB, M., ZARFOSS, J., MONROSE, F., AND TERZIS, A. 2006. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the Internet Measurement Conference*. P. Barford Ed., ACM, New York, NY, 41–52.

RAYMOND, J.-F. 2000. Traffic analysis: Protocols, attacks, design issues, and open problems. In *Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability*. 10–29.

REITER, M. AND RUBIN, A. 1998. Crowds: Anonymity for Web transactions. *ACM Trans. Inf. Syst. Sec. 1*, 1, 66–92.

RENNHARD, M. AND PLATTNER, B. 2002. Introducing MorphMix: Peer-to-peer based anonymous Internet usage with collusion detection. In *Proceedings of the Workshop on Privacy in Electronic Society*. ACM, New York, NY, 91–102.

ROWSTRON, A. AND DRUSCHEL, P. 2001. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*. G. Goos, J. Hartmanis, and J. van Leeuwen Eds., Lecture Notes in Computer Science, vol. 2218. Springer, Berlin, 329–350.

SERJANTOV, A. AND DANEZIS, G. 2002. Towards an information theoretic metric for anonymity. In *Proceedings of the Workshop on Privacy Enhancing Techonologies*. 259–263.

SHERR, M., LOO, B. T., AND BLAZE, M. 2007. Towards application-aware anonymous routing. In *Proceedings of the 2nd USENIX Workshop on Hot Topics in Security*. USENIX Association, Berkeley, CA, 4:1–4:5.

SIT, E. AND MORRIS, R. 2002. Security considerations for peer-to-peer distributed hash tables. In *Proceedings of the 1st International Workshop on Peer-to-Peer System*. 261–269.

STOICA, I., MORRIS, R., LIBEN-NOWELL, D., KARGER, D. R., KAASHOEK, M. F., DABEK, F., AND BALAKRISHNAN, H. 2003. Chord: A scalable peer-to-peer lookup protocol for Internet applications. *IEEE/ACM Trans. Netw. 11*, 1, 17–32.

SYVERSON, P., TSUDIK, G., REED, M., AND LANDWEHR, C. 2000. Towards an analysis of onion routing security. In *Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability*. 96–114.

TABRIZ, P. AND BORISOV, N. 2006. Breaking the collusion detection mechanism of MorphMix. In *Proceedings of the 6th Workshop on Privacy Enhancing Techonologies*. 368–383.

THE TOR PROJECT. Tor metrics portal, http://metrics.torproject.org/ (last accessed 2/11).

WALLACH, D. 2002. A survey of peer-to-peer security issues. In *Proceedings of the International Symposium on Software Security*. M. Okada, B. Pierce, A. Scedrov, H. Tokuda, and A. Yonezawa Eds., Lecture Notes in Computer Science, vol. 2609. Springer, Berlin, 253–258.

WANG, Q., MITTAL, P., AND BORISOV, N. 2010. In search of an anonymous and secure lookup: Attacks on structured peer-to-peer anonymous communication systems. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS'10)*. A. D. Keromytis and V. Shmatikov Eds., ACM.

WRIGHT, M., ADLER, M., LEVINE, B. N., AND SHIELDS, C. 2002. An analysis of the degradation of anonymous protocols. In *Proceedings of the Network and Distributed System Security Symposium*. P. van Oorschot and V. Gligor Eds., The Internet Society, Reston, VA, 39–50.

WRIGHT, M., ADLER, M., LEVINE, B. N., AND SHIELDS, C. 2003. Defending anonymous communication against passive logging attacks. In *Proceedings of the IEEE Symposium on Security and Privacy*. 28–41.

WRIGHT, M., ADLER, M., LEVINE, B. N., AND SHIELDS, C. 2004. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Trans. Inf. Syst. Secur. 4*, 7, 489–522.

WRIGHT, R. AND DI VIMERCATI, S. D. C., Eds. 2006. In *Proceedings of the The 13th ACM Conference on Computer and Communications Security*. ACM, New York, NY.

WRIGHT, R. AND SYVERSON, P., Eds. 2007. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*. ACM, New York, NY.

ZETTER, K. 2010. Wikileaks and Tor. http://www.wired.com/threatlevel/2010/06/wikileaks-documents/.

Received March 2009; revised February 2011; accepted June 2011