A SECURITY EVALUATION OF THE SALSA ANONYMOUS COMMUNICATION SYSTEM

BY

PRATEEK MITTAL

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Adviser:

Assistant Professor Nikita Borisov

ABSTRACT

We evaluate a state of the art P2P anonymous communication system, Salsa. Salsa is based on a distributed hash table, and uses secure lookups to locate relays for anonymous communication. To analyze user anonymity in Salsa, we first build an analytic model for the lookup security in Salsa, and model its path building mechanism as a stochastic activity network in the Möbius framework.

Next, we analyze information leaks in the lookup mechanisms of Salsa and show how these leaks can be used to compromise anonymity. We show that the techniques that are used to combat active attacks on the lookup mechanism dramatically increase information leaks and increase the efficacy of passive attacks. Thus there is a tradeoff between active and passive attacks. We find that, by combining both passive and active attacks, anonymity can be compromised much more effectively than previously thought.

We also show that Salsa is vulnerable to a selective DoS attack, where an adversary denies service whenever he/she is unable to compromise user anonymity. This attack is devastating for user anonymity in Salsa, rendering the system insecure for most proposed uses. Finally, we perform a first step towards an entropy based evaluation of Salsa, instead of considering the binary metric of path compromise, which results in an even lower user anonymity. Our study therefore motivates the search for new approaches to P2P anonymous communication.

To my family

ACKNOWLEDGMENTS

First of all, I am very grateful to my adviser, Nikita Borisov, whose continued guidance, encouragement, and support has shaped this thesis. I would also like to thank my fellow students and faculty at UIUC for their helpful comments and feedback. In particular, I am grateful to my colleagues in the Hatswitch research group - Amir Houmansadr, Robin Snader, David Albrecht, Nabil Schear and Shishir Nagaraja for many insightful discussions.

Outside of Illinois, I am thankful to George Danezis, Matthew Wright, Parisa Tabriz, and Apu Kapadia for their invaluable inputs on my research.

Chapter 4 and Chapter 5, in part, have been published at the 15th ACM Conference on Computer and Communication Security 2008, and the 14th ACM Conference on Computer and Communication Security 2007 respectively. Elements from Chapter 3, Chapter 4, and Chapter 6 are currently under submission for the ACM Transactions on Information and System Security.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Anonymous communication hides the identity of communication partners from third parties, or hides user identity from the remote party. The Tor network [1], deployed in 2003, now serves hundreds of thousands of users and carries terabytes of traffic a day [2]. Originally an experimental network used by privacy enthusiasts, it is now entering mainstream use; for example, several consulates were found to be using it to evade observation by their host country [3].

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 [4, 5, 6, 7]. Salsa is the state of art in peer-to-peer anonymous communication systems, and the subject of this thesis. We present an overview of Salsa in Chapter 2, describing its lookup and path building mechanisms. We also discuss its threat model, in addition to motivating the need for scalable approaches to anonymous communication.

Prior work on analyzing Salsa used the help of simulations to analyze the lookup security and user anonymity. Our first contribution in this thesis is a theoretical analysis of Salsa. We present an analytic model for lookup security in Salsa, as well as a stochastic activity network to model anonymous path construction in Salsa. We provide a description of the theoretical model in Chapter 3.

A key challenge in peer-to-peer anonymous communication systems is the ability to locate relays for anonymous traffic. 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. 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; e.g. AP3 [4] uses secure routing techniques developed by Castro et al [8], and Salsa uses redundant routing with bounds checks [5].

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. As most modern anonymity systems assume that a global passive adversary is too costly, they are not designed to resist such attacks. Therefore, this small fraction of attackers can successfully attack anonymity of the system.

Our next contribution is the analysis of such information leak attacks in Salsa. We find that defenses against active attacks create new opportunities for passive attacks. Salsa makes heavy use of redundancy to address active attacks, rendering it vulnerable to passive information leak attacks. Further, increasing the levels of redundancy will improve passive attack performance, and 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 paths will be compromised by using information leaks. We also 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. We discuss and analyze these information leak attacks in Chapter 4.

Next, we consider a selective denial-of-service attack on Salsa. Instead of blanket denial-of-service attack, an adversary may *selectively* affect reliability of the system in states that are hardest to compromise, thereby causing the system to enter less secure states. In particular, we explore an attack where DoS is performed whenever communication cannot be compromised. Such selective DoS is both easier to carry out than an attack on the entire system, and can be more effective; instead of driving the users away from the system, selective DoS presents them with a less reliable, but still functional system. Faced with poor reliability, many users will naturally attempt the communication again, presenting more opportunities for attack. In Chapter 5, we show that the selective DoS attack is devastating for user anonymity in Salsa; at 20% compromised nodes, the probability of path compromise is 0.7, thus rendering the system insecure for most proposed uses.

Conventional anonymity analysis of Salsa considers the binary metric of path compromise. Even our model for information leak attacks discussed earlier is restricted to the scenario where the adversary is able to precisely identify the initiator of a lookup. In Chapter 6, we extend our model by performing a first step towards an entropy based evaluation of Salsa, that considers a distribution of possible initiators of a lookup. We show that an entropy based model for information leaks results in an even lower user anonymity. Our results demonstrate that information leaks are an important part of anonymity analysis of a system and that new advances in the state of the art of P2P anonymous communication are needed. Finally, we discuss the related work in Chapter 7, and conclude in Chapter 8.

CHAPTER 2

BACKGROUND

In this chapter, 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 then describe our threat model, as well as the design of Salsa.

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 [9] and Mixmaster [10] 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 paper is on low-latency anonymous communication systems.

Tor [1] 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 [11]. 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 [6] 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 [7] 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. In MorphMix, a node along the path is queried for its neighbors in order to choose the next hop. To prevent the 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 [12].

Several other designs have used so-called structured peer-to-peer topologies [4, 5], 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. Salsa [5] is the state of art design, which aims to offer secure P2P anonymous communication in a system without a PKI. Its design includes 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, presenting a trade-off between resisting active attacks and presenting more opportunities for passive attacks, as well as makes Salsa more vulnerable to a selective DoS attack.

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 to be static and the adversary cannot compromise nodes at will. In terms of the standard terminology introduced by [13], 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. Most peer-to-peer systems including Salsa use mechanisms to prevent Sybil attacks [14], which would 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 is often measured in tens of thousands of nodes [15, 16], present a very real threat to anonymity.

2.3 Salsa

Salsa [5] is an anonymous communication system designed to overcome the scalability problems in traditional mix systems. As in Tor, a circuit is built between the initiator and the recipient via proxy routers (nodes) for anonymous communication. Layered encryption ensures that each node knows only its previous and next hop in the circuit. The nodes used for the circuits are randomly selected from the global pool of nodes, even though each node has only local knowledge of a small subset of the network.

2.3.1 Salsa Architecture

Salsa is based on a distributed hash table (DHT) that maps nodes to a point in an ID space corresponding to the hash of their IP address. The ID space in Salsa is divided into equal sized groups, organized into a binary tree structure. Each node knows all the nodes in its group (local contacts), and a small number of nodes nodes in other groups (global contacts). Each node maintains one global contact for every level of the binary tree. At every level, the global contact is selected at random from the subtree corresponding to the other child of the node's parent at that level.

There are two basic mechanisms in Salsa: (1) a node lookup mechanism and (2) a circuit building mechanism. The former returns the IP address and public key of node in the DHT closest to a given point in the ID space. The latter is used to build a Tor-like tunnel.

2.3.2 Salsa Secure Lookup

Similar to Pastry, nodes must rely on other nodes to perform a recursive lookup. The initiator of the lookup contacts its global contact in the same subtree as the destination identifier to continue the lookup. The lookup proceeds in a recursive fashion until the destination identifier is in the same subgroup as the intermediate requesting node; in this case, the intermediate requesting node can simply return the IP address and public key of the closest node to the destination identifier.

A malicious node who 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 binary tree architecture is designed to ensure that redundant paths have very few common nodes between them (unlike Pastry or Chord [17]). 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 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.

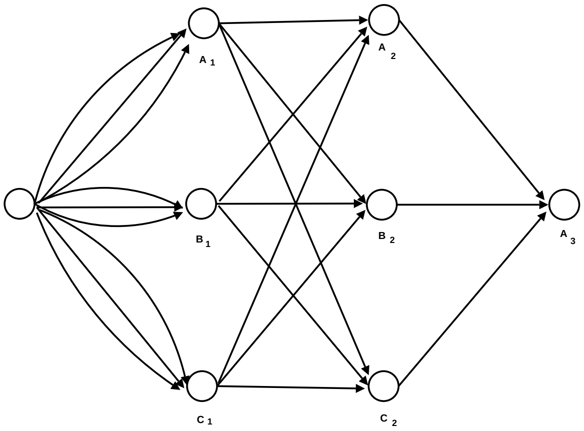


Figure 2.1: Salsa path construction.

2.3.3 Salsa Circuit Construction

To build a circuit, the initiator chooses r random IDs ([5] sets $r = 3$) 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 ([5] sets $l = 3$). Figure 2.1 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.

CHAPTER 3

AN ANALYTIC MODEL FOR SALSA

3.1 Analytic Model for Lookup

We denote the initiator of the lookup as I, and the target identifier as ID. Let us consider the following two possibilities, as depicted in Figure 3.1. In the first scenario, the node corresponding to ID is malicious, which happens with probability f. If this malicious node passes the bounds check (with probability $1 - \Delta_1$), the resulting lookup is compromised. If this malicious node fails the bounds check (with probability Δ_1), the lookup is aborted and I performs a lookup for some other identifier. In the second scenario, the node corresponding to ID is honest. The following cases are possible in this scenario: (a) There is at least one lookup path with all honest nodes (the probability of which is denoted by g). Now if the honest node corresponding to ID succeeds the bounds check (with probability $1 - \Delta_1$), the lookup is successful, else if the honest node fails the bounds check (with probability Δ_1), the lookup is aborted. (b) Every lookup path has at least one malicious node (with probability $1 - g$). Now, if there is a malicious node within bounds (with probability Δ_2), the resulting lookup is compromised, otherwise the lookup is aborted.

Δ_1 is the probability of false positives: i.e. there is no node with an identifier in the range between between target ID and $ID + b$, where b is the bounds check parameter. If we consider the ID space to be of unit size, then Δ_1 can be computed as

$$\Delta_1 = (1 - b)^N \tag{3.1}$$

Δ_2 is the probability of a false negative: i.e. given that the target node is honest, there is a malicious node within bounds. Suppose that the target

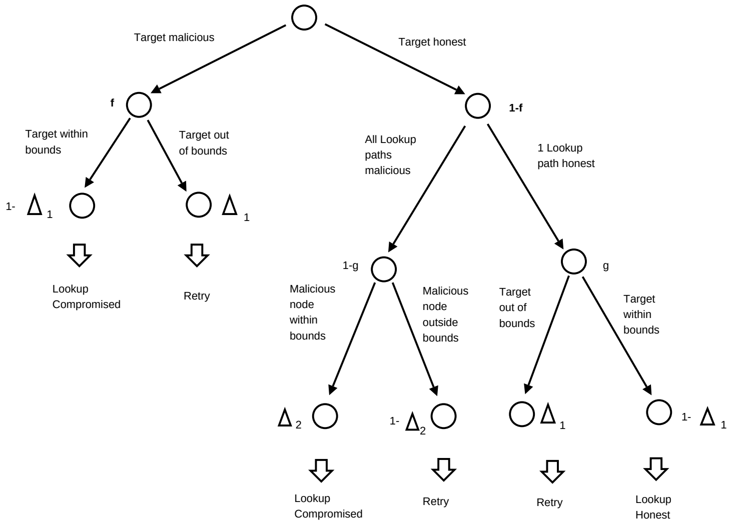


Figure 3.1: Computing probability of compromised lookup.

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 that

$$\Delta_2 = P(\text{malicious node within bounds}|\text{ target node is honest}) \tag{3.2a}$$

$$\Delta_2 = 1 - P(\text{malicious node outside bounds}|\text{target node is honest}) \tag{3.2b}$$

$$\Delta_2 = 1 - \int_{a=0}^{b} f(a) \cdot (\frac{1-b}{1-a})^{N \cdot f}\, da. - \int_{a=b}^{1} f(a) \cdot 1\, da. \tag{3.2c}$$

$$\Delta_2 = 1 - \int_{a=0}^{b} N \cdot (1-a)^{N-1} \cdot (\frac{1-b}{1-a})^{N \cdot f}\, da. - \int_{a=b}^{1} N \cdot (1-a)^{N-1}\, da. \tag{3.2d}$$

$$\Delta_2 = 1 - N \cdot (1-b)^{N \cdot f} \cdot \frac{1-(1-b)^{N - N \cdot f}}{N - N \cdot f} - \Delta_1 \tag{3.2e}$$

The term 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.

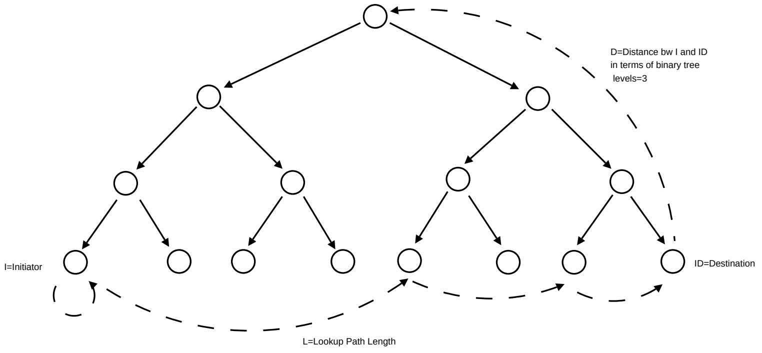


Figure 3.2: Salsa binary tree structure.

3.1.1 Single Lookup, $r = 1$

Let us denote the lookup path length by L. Given a particular lookup path length $(L = l)$, we have that

$$g = (1 - f)^l \tag{3.3}$$

Based on the Figure 3.1, we have that

$$\begin{aligned} &P(\text{Compromised Lookup}|L = l) = \\ &\qquad \frac{f \cdot (1 - \Delta_1) + (1 - f) \cdot (1 - g) \cdot \Delta_2}{f \cdot (1 - \Delta_1) + (1 - f) \cdot (1 - g) \cdot \Delta_2 + (1 - f) \cdot g \cdot (1 - \Delta_1)} \end{aligned} \tag{3.4}$$

where Δ_1, Δ_2, g have been computed in Equations (3.1), (3.2) and (3.3).

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 3.2. In order to compute $P(L = l)$, we can first condition on the event $D = d$. Since I selects the target ID uniformly at random from the ID space, the probability that the target is d levels away from the initiator in the binary tree structure is

$$P(D = d) = \begin{cases} \frac{2^{d-1}}{G} & d \geq 1 \\ \frac{1}{G} & d = 0 \end{cases} \tag{3.5}$$

Under the event $D = d$, we shall compute the probability of lookup path length being l hops, i.e. $P(L = l|D = d)$. 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, where 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 that

$$P(L = l|D = d) = \begin{cases} \binom{d-1}{l-2}(\frac{1}{2})^{d-1} & d \geq 1 \\ 1 & d = 0, l = 1 \\ 0 & d = 0, l > 1 \end{cases} \tag{3.6}$$

Using Equations (3.5) and (3.6), we can compute $P(L = l)$ as follows:

$$P(L = l) = \sum_{d=0}^{\log G} P(L = l|D = d) \cdot P(D = d) \tag{3.7a}$$

$$P(L = l) = \begin{cases} \sum_{d=1}^{\log G} \binom{d-1}{l-2} \cdot \frac{1}{G} & l \geq 2 \\ \frac{1}{G} & l = 1 \end{cases} \tag{3.7b}$$

Finally, using Equations (3.4) and (3.7) we can compute the probability of a compromised lookup as follows:

$$P(\text{Compromised Lookup}) = \sum_{l=1}^{\log G+1} P(\text{Compromised Lookup}|L = l) \cdot P(L = l) \tag{3.8}$$

3.1.2 Redundant Lookups

Let us denote the r lookup path lengths by $L_1, L_2 ... L_r$. Given particular lookup path lengths $(L_1 = l_1 ... L_r = l_r)$, we have that

$$g = P(\text{at least one lookup path is honest}) \tag{3.9a}$$
$$g = 1 - P(\text{all lookup paths have a malicious node}) \tag{3.9b}$$
$$g = 1 - \prod_{j=1}^{r} 1 - (1-f)^{l_j} \tag{3.9c}$$

Based on the Figure 3.1, we have that

$$P(\text{Compromised Lookup}|L_1 = l_1 .. L_r = l_r) = \\ \frac{f \cdot (1-\Delta_1) + (1-f) \cdot (1-g) \cdot \Delta_2}{f \cdot (1-\Delta_1) + (1-f) \cdot (1-g) \cdot \Delta_2 + (1-f) \cdot g \cdot (1-\Delta_1)} \tag{3.10}$$

where Δ_1, Δ_2, g have been computed in equations (3.1), (3.2) and (3.9). Now we shall compute $P(L_1 = l_1 .. L_r = l_r)$ by conditioning on the event $D = d$. Note that conditioned on $D = d$, the redundant lookups are independent. Thus, we have that

$$P(L_1 = l_1 .. L_r = l_r | D = d) = \prod_{j=1}^{r} P(L_j = l_j | D = d) \tag{3.11}$$

Using Equation (3.11), we can compute $P(L_1 = l_1 .. L_r = l_r)$ as follows :

$$P(L_1 = l_1 .. L_r = l_r) = \sum_{d=0}^{\log G} P(L_1 = l_1 .. L_r = l_r | D = d) \cdot P(D = d) \tag{3.12a}$$
$$P(L_1 = l_1 .. L_r = l_r) = \sum_{d=0}^{\log G} (\prod_{j=1}^{r} P(L_j = l_j | D = d)) \cdot P(D = d) \tag{3.12b}$$

where $P(L = l | D = d)$ and $P(D = d)$ are given by Equations (3.6) and (3.5). Finally, using Equations (3.10) and (3.12) we can compute the probability of a compromised lookup as follows:

$$P(\text{Compromised Lookup}) =$$
$$\sum_{l_1=1}^{\log G+1} .. \sum_{l_r=1}^{\log G+1} P(\text{Compromised Lookup}|L_1 = l_1..L_r = l_r) \cdot P(L_1 = l_1..L_r = l_r) \tag{3.13}$$

3.2 Analytic Model for Circuit Construction

The path construction mechanism in Salsa is quite complex, and difficult to model by hand. Instead, we will model it as a stochastic activity network (SAN) using the Möbius framework [18].

3.2.1 Möbius Framework

Möbius is a multi-formalism, multi-solution framework for computer systems analysis. While originally designed for systems level performance analysis (reliability, availability), its flexibility has enabled its application to a wide range of discrete event systems including modeling attacks on secure systems. The main components of Möbius are as follows:

- Atomic Models: Möbius supports stochastic extensions to Petri-Nets, Markov chains and extensions, and stochastic process algebras. We will use the formalism of stochastic activity network to describe the path construction in Salsa.

- Reward Variables: Reward variables allow for detailed customized measurement of the system properties, including periodic measurements and measurements at the steady state. Our approach is to define a state in the SAN model indicating the compromise of user anonymity, and then measure the steady state properties of that state.

- Study: The study component allows us to define input parameters to the model, and then study the behavior of the system over a wide range of input parameter values. We use the number of nodes in each stage of path building (r), and the number of stages (l) as inputs to the analytic model. This enables us to compute anonymity over different possible choices of r, l in Salsa.

- Solver: Möbius allows for both distributed discrete event simulation as well numerical solution techniques. We will use the numerical solution technique to solve the SAN model, because it is able to compute *exact* solutions to models with tens of millions of states.

We shall now describe the SAN model for Salsa path construction.

3.2.2 Stochastic Activity Network (SAN) Model for Path Construction

Stochastic activity networks are a convenient, graphical, high level language for describing system behavior. SANs consist of the following:

- Place: places are like variables, and contain *tokens*, which are the value of the place (variable).

- Transition and Cases: transitions change the value of the state, and cases are used to specify probabilistic choices.

- Input Gates: connect states to transitions, and are used to define complex enabling and completion functions.

- Output Gates: connect transitions to states, and are used to define complex completion functions.

Figure 3.3 depicts the SAN model for path construction in Salsa. The *state* place is a complex data structure comprising a Boolean variable for all nodes in the path building process $(r \cdot (l-1)+1)$ and two integer variables for recording the current node (denoted by q, s.t. $1 \leq q \leq r \cdot (l-1)+1$)and the current stage (denoted by k, s.t. $1 \leq k \leq l$) of the process. The Boolean variables indicate whether the selected nodes are honest or malicious. The current stage of the process is initialized as $q = 0, k = 0$.

The input gate ig is enabled as long as $k \leq l-1$. If the input gate ig is enabled, then the transition *set_malicious* computes the probabilities of the nodes in the next stage being malicious, based on the number of honest

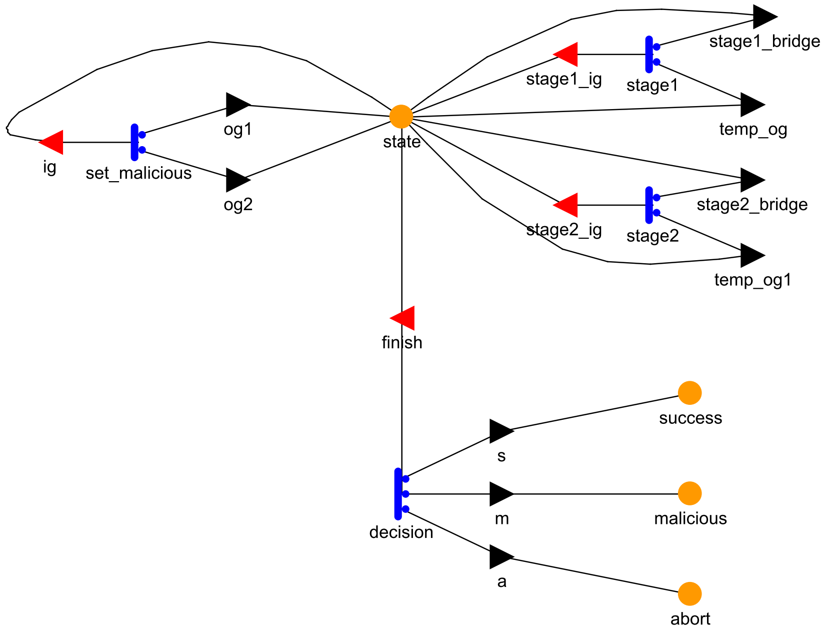


Figure 3.3: Stochastic activity network for path construction in Salsa.

nodes in the previous stage. Notice that if x nodes in the $k'th$ stage are malicious, then effectively, only $r - x$ nodes are performing lookups for the nodes in the next stage $k + 1$, and we can compute the probabilities of nodes in the next stage being honest using the analytic model for lookup developed earlier and setting the number of redundant lookups to $r - x$. Note that for the first stage, we set $x = 0$, as all the lookups are performed by the initiator.

Suppose the *set_malicious* function computes the probability of nodes in the next stage being honest as p, then with probability p, output gate $og1$ is chosen and it sets the Boolean variable corresponding to the current node in the current stage as honest. With probability $1 - p$, output gate $og2$ is chosen, and it sets the Boolean variable corresponding to the current node in the current stage as malicious. Both output gates $og1$ and $og2$ also increment the current value of the current node (q). Furthermore if $q\%r == 0$, then the value of the current stage is incremented $(k + +)$. Note that the last stage is an exception as only one node needs to be selected.

When the value of k reaches l, the input gate ig is disabled, and the node selection procedure is complete. Using the power of cases in transitions, we have been able to model the cascading effect in the Salsa path building process, i.e., the choice of nodes in a stage affects the choice of nodes in the next stage.

The input gate $finish$ is enabled when the value of k is equal to l. The transition $decision$ encodes the attacker's algorithm once the node selection process is finished, e.g., passive timing analysis attacks. If the attack is successful, the output gate m is selected which increments the value of the $malicious$ place (which acts as an absorbing state). Otherwise, the output gate s is selected which increments the value of the $success$ place. The other modules in the Figure 3.3 correspond to attacks discussed in the future chapters. The modules corresponding to $stage1$ and $stage2$ are used for passive bridging attacks described in Chapter 4, while the $abort$ module is used for the selective denial of service attack described in Chapter 5. We set the reward variables in the Möbius framework to be the values of the places $malicious$ and $success$, and compute the final probability of user compromise as $\frac{malicious}{malicious+success}$.

CHAPTER 4

INFORMATION LEAKS ATTACKS ON SALSA

4.1 Information Leaks via Secure Lookups

It has been recognized that unprotected DHTs are extremely vulnerable to attacks on the lookup mechanism. First of all, malicious nodes can perform a Sybil attack [14] 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, 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.

In Chapter 3, we have seen how 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 [17]). 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.

To validate our mathematical model of Salsa lookup, we used a simulator developed by the authors of Salsa [19].[1] The simulator was configured to simulate 1000 topologies, and in each topology, results were averaged over 1000 random lookups. The lookup bias is sensitive to the average lookup path length, which in turn is 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 1000 nodes in our simulation.

The choice of the parameter b has an interesting tradeoff. Decreasing the bounds checking distance will increase the probability that a legitimate root of the key lies outside bounds. This scenario is a false positive. An increase in b would lead to a reduction in false positives, but then attackers could have some nodes that are within the bounds of a random key, even if they are not the root (false negatives). Thus if we decrease the false positives by increasing the bounds checking distance, the false negatives will increase. A practical strategy is to keep the false positives small by having a relatively higher b, and to reduce the false negatives by making use of diverse paths (redundant routing). Salsa sets the bounds checking distance as $b = \mathit{offset} \cdot \mathit{groupsize}$; the corresponding false positives in bounds checking can be computed as $(1-b)^N$. For 128 groups and 1000 nodes, using an offset value of 0.5 results in less than 2% false positives. We shall use this value in the remainder of our analysis. Note that the false positives should be small for performance reasons, else the lookup initiator would have to perform many lookups to get a root which is within bounds.

Salsa is resistant to conventional attacks that target the lookup mechanism as long as the fraction of malicious nodes in the system is less that 20%. Since Salsa does not provide adequate security for higher values of f, we shall limit our analysis to low values. In Figure 4.1, we study the effect of varying redundancy on the lookup bias. First, we note that the simulation estimates closely match our analytic results. Second, we can see that increasing r clearly reduces the fraction of compromised lookups, thus increasing security. For $f = 0.2$, the fraction of compromised lookups drops

[1] Our results differ slightly from those shown in [5] because of a bug in the simulator. We have communicated the bug to the authors and it has been accepted.

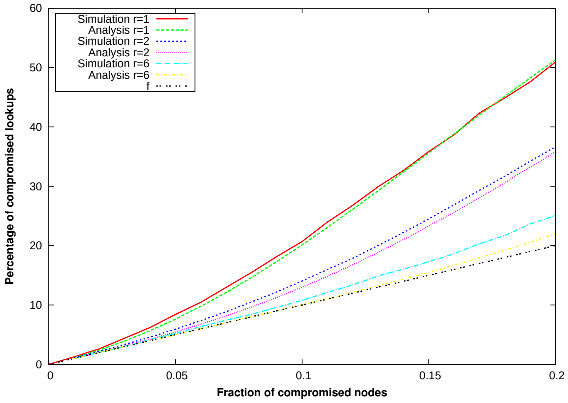


Figure 4.1: Percentage of compromised lookups.

from 37% to 25% when r is increased from 2 to 6.

However, the secure lookup mechanism generates many extra messages: 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. In particular, the initiator of a lookup can be precisely 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 4.2. Clearly, increasing r increases the chance that a lookup initiator is detected. This illustrates the trade-off between security and anonymity of a lookup. We also note that information leaks are inherent in other secure routing protocols as well, like that of Castro et al. [8]. In fact, for Castro et al. [8], when only 5% nodes are malicious, they observe more than 60% of all lookups. AP3 is an anonymity system based on the secure lookup mechansim of Castro et al. [8], and we have analyzed its security in the Appendix.

This shows the fundamental tension that is encountered by a DHT lookup. The default DHT 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. Salsa's secure routing 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

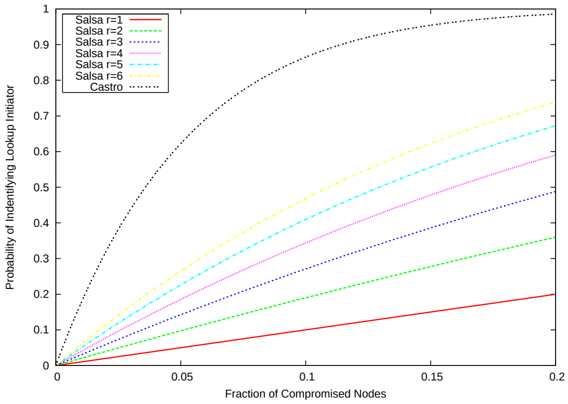


Figure 4.2: Information leaks from secure lookups.

than anonymous communication that are built on top of secure lookup. So far, we have observed a tradeoff between security and anonymity of a lookup; we shall now use this to break the user anonymity in Salsa.

4.2 Attacks on Salsa Path Construction

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.

4.2.1 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. [20] noted that Salsa path building is also subject to a public key modification attack.[2] If all the nodes in a particular stage are compromised, they can modify the public keys of the next set of nodes being looked up. This attack defeats Salsa's 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

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

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. Now, 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. This is because 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 shall also show that increasing redundancy increases the effectiveness of the information leak attacks, resulting in a trade-off between robustness against active attacks and passive information leak attacks.

4.2.2 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 in [5]. 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 shall describe two new bridging attacks also based on information leaks from lookups.

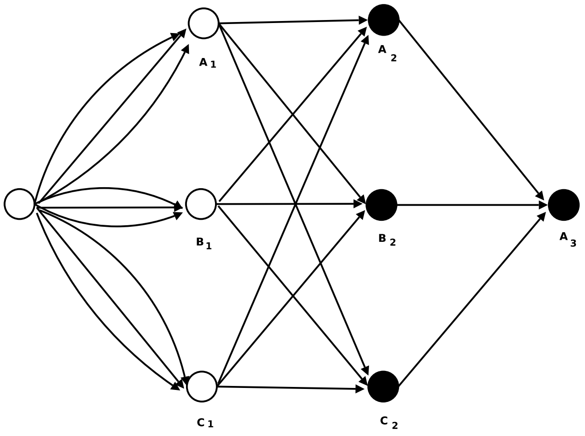


Figure 4.3: Bridging an honest first stage.

4.2.3 Bridging an Honest First Stage

This attack is based on the observation that the 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 detect any of the initiator's redundant lookups for nodes in the first stage, the anonymity of the system is compromised. Consider Figure 4.3; 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, C_1 because of key establishment with them. Now 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 9 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.

It is important to note that there are some false positives in the attack.

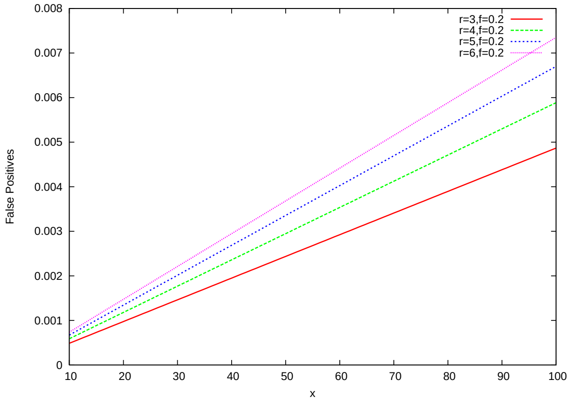


Figure 4.4: False positives in bridging an honest first stage.

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 look up A_1, and the attackers may detect a lookup for A_1 not done by the actual initiator. 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. Using the variable x to model the amount of lookup traffic by other nodes, we can compute the false positives as

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

Figure 4.4 depicts the false positives for varying r using $f = 0.2$ and $N = 1000$. Note that for $x < \frac{N}{100}$, the false positives are less than 0.1%.

4.2.4 Bridging an Honest Stage

Salsa is also vulnerable to a bridging attack where 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 4.5. 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 $1 - (\frac{N-1}{N})^x$, which for $x = N/100$ and $N = 1000$ is less than 1%.

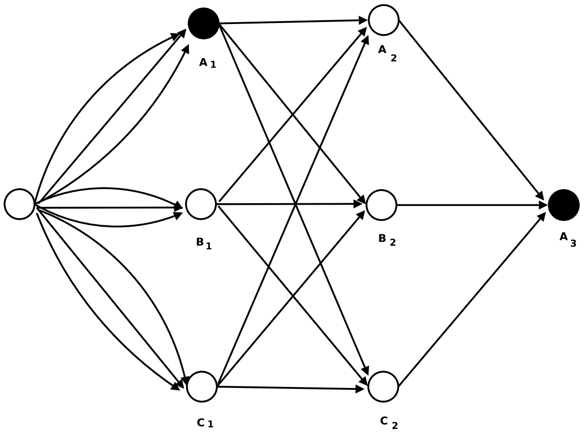


Figure 4.5: Bridging an honest stage.

4.2.5 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 [18]. The input to the model is the lookup bias, computed using our analytic model, as described in Chapter 3.

Figure 4.6 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

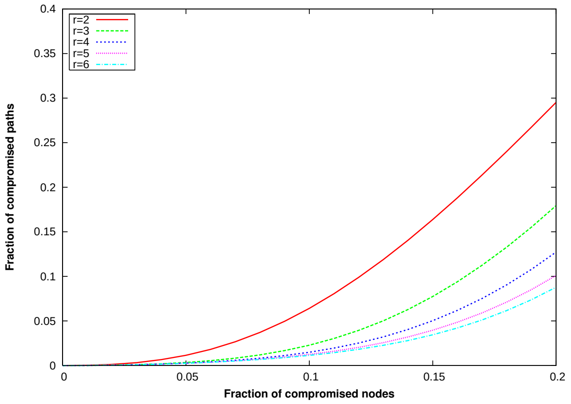


Figure 4.6: Conventional path compromise attacks: Increasing redundancy counters active attacks.

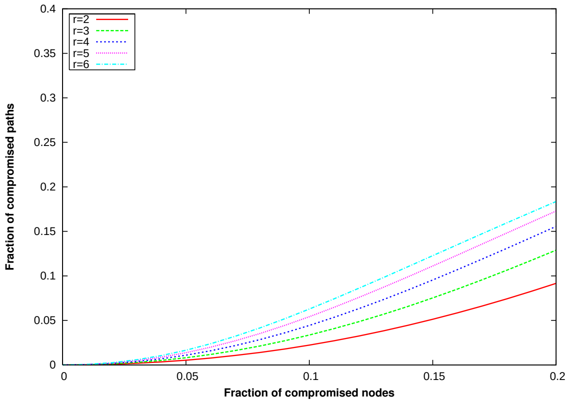


Figure 4.7: Information leak attacks: Increasing redundancy makes the passive adversary stronger.

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 4.7 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% 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 4.8 shows the fraction of compromised paths for all passive and active attacks. An interesting trend is observed where increasing redundancy (beyond $r = 2$) is detrimental to security for small values of f. This is in sharp contrast to 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 to 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.

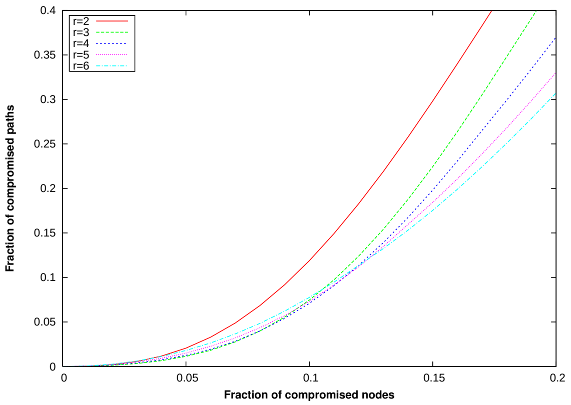


Figure 4.8: All conventional and information leak attacks: For maximal anonymity, $r = 3$ is optimal for small f. Note that there is a crossover point at $f = 0.1$ when $r = 6$ becomes optimal.

This shows the tension between the passive and active attacks. There is an inherent redundancy in the Salsa path building mechanism to counter active attacks. However, the redundancy makes the passive adversary stronger and provides more opportunities for attack. From Figure 4.9 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.

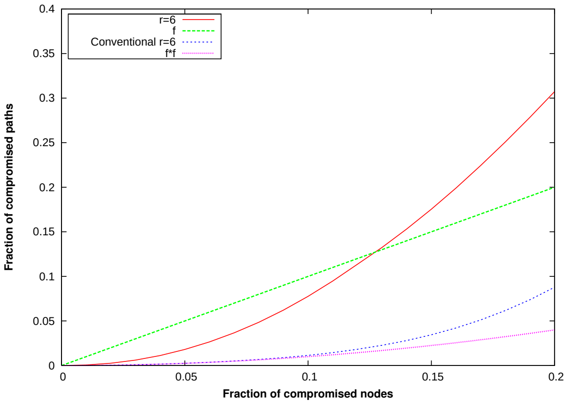


Figure 4.9: Comparison of all attacks with conventional active attacks: Note that for $f > 0.12$, fraction of compromised paths is greater than f.

4.2.6 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 4.10 depicts the probability of identifying the initiator under all active and passive attacks in Salsa with PKI. Again, we can see the tension between active and passive attacks. Increasing redundancy (beyond $r = 2$) is detrimental to security for small values of f, because of the dominance of our information leak attacks. There is a crossover point, after which active attacks become dominant, and increasing r increases security. 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 4.11 depicts the probability of identifying the initiator for varying values of l. There is an interesting trade-off 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 last node is

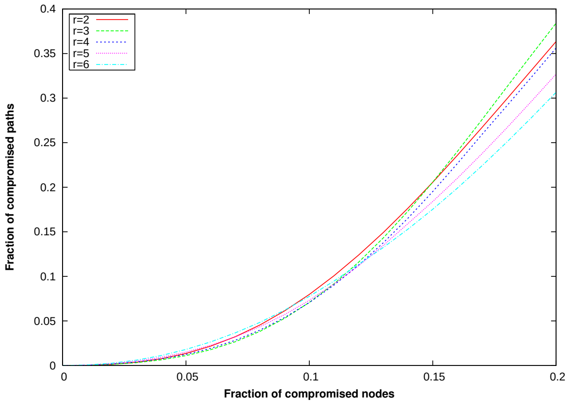


Figure 4.10: Salsa with a PKI—All conventional and information leak attacks. Even with a PKI, the security of Salsa is much worse as compared to conventional analysis.

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 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 are dominant, therefore increasing l decreases security.

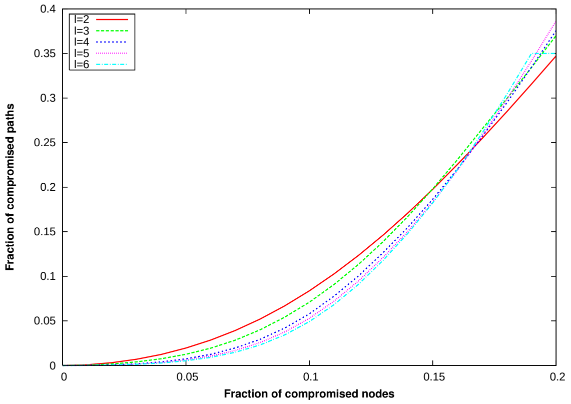


Figure 4.11: Effect of varying the path length: Note that there is only limited benefit of increasing path length.

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 trade-off between defenses against active and passive attacks. Even at the optimal point in the trade-off, the anonymity provided by the system is significantly worse than what was previously thought. This trade-off is present even in Salsa with a PKI. Moreover, increasing path length in Salsa has only a limited benefit for the user anonymity.

CHAPTER 5

DENIAL OF SERVICE AGAINST SALSA

5.1 Selective DoS Attack

We now consider a selective denial-of-service attack on Salsa. Instead of blanket denial-of-service attack, an adversary may *selectively* affect reliability of the system in states that are hardest to compromise, thereby causing the system to enter less secure states. The idea of selective DoS attack is to deny service to trustworthy nodes so that user traffic moves toward compromised nodes. The compromised nodes will try to abort the tunnel building process whenever the tunnel cannot be compromised. A malicious node can easily launch a denial of service by returning an arbitrary result from a lookup. The Salsa tunnel building mechanism aborts if the lookup information provided by the redundant r nodes in any stage is inconsistent.[1] Such selective DoS is both easier to carry out than an attack on the entire system, and can be more effective; instead of driving the users away from the system, they are presented with a less reliable, but still functional system. Faced with poor reliability, many users will naturally attempt the communication again, presenting more opportunities for attack.

The attackers should deny service in two cases. First, if the last node is honest, and there is an attacker in the second last stage, that attacker will perform DoS, unless all r nodes in that stage are malicious. (This can be easily determined on the reception of r messages at attacker nodes containing lookup requests for the identical r nodes in the next stage.) Also, if the attacker nodes are selected to forward traffic in a tunnel, they can deny service if the tunnel has not been compromised. The nodes will perform traffic analysis on the first portion of the stream sent over a tunnel

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

and correlate it with all other streams observed by other attackers. If the stream can be linked to both an initiator and a destination, the attackers continue forwarding traffic; otherwise, they terminate the tunnel as it cannot be compromised.

The attack algorithm is as follows:

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

5.2 Analysis

We compare the performance of three attack methodologies on the Salsa tunnel building mechanism. The first one consists of conventional active attacks on lookups, our public key modification attack and end-to-end timing analysis. The second methodology involves using the passive information leak attacks, in addition to conventional active attacks. In the third methodology, nodes try to selectively DoS the tunnels which are likely not to be compromised. All other attacks are also included in this methodology. In our analysis, we have assumed that a user strives for perfect reliability. Our results have been computed by modeling the Salsa tunnel building mechanism as a stochastic activity network in the Möbius framework [18] as illustrated in Chapter 3. Figure 5.1 shows the fraction of compromised tunnels for varying attacker ratios under the three attacks.

Our analysis shows that the current Salsa design is extremely vulnerable to the selective DoS attack, especially for high attacker ratios. In fact, as compared to the our own security analysis of 39.2% compromised tunnels for an attacker ratio of $f = 0.2$ (the second attack methodology), the selective DoS attack results in 71% compromised tunnels. Also, a majority of all tunnels are compromised when $f \geq 0.17$. This shows that the selective DoS attack has devastating effects on the security of Salsa.

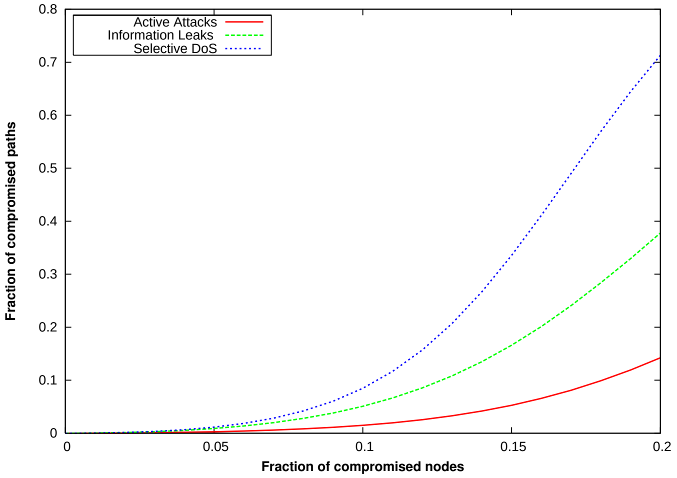


Figure 5.1: Effect of selective DoS on Salsa tunnel building.

Given the massive reduction in anonymity made possible by the selective DoS attack, we study whether other choices of r and l could better resist this attack. We find that an increase in the number of nodes in a stage (r) or the number of stages (l) does not improve system anonymity under selective DoS.

Figure 4.8 shows the effect of varying r under information leak attacks. We can see that for small values of f, passive information leak attacks dominate and increasing redundancy increases the the fraction of compromised tunnels. There is a crossover point at about $f = 0.1$, when active attacks begin to dominate, and increasing redundancy reduces the fraction of compromised tunnels. Figure 5.2 shows the effect of varying r on the system anonymity under selective DoS attack. Again, we can see that for small values of f, the passive information leak attacks dominate and increasing redundancy reduces anonymity. Now, there is an interesting

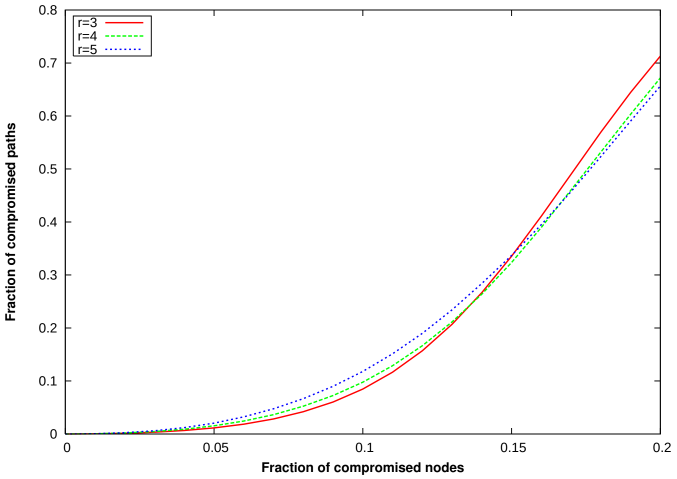


Figure 5.2: Effect of varying r.

tradeoff between selective DoS attack and active attacks on lookup. Increasing redundancy mitigates conventional active attacks, but gives more opportunities to the attackers to launch selective DoS. Because of the selective DoS attack, the crossover point where increasing redundancy is beneficial for anonymity has shifted to about $f = 0.15$. Observe that even for $f \geq 0.15$, the advantage offered by increasing redundancy is very small (not worth the communication overhead of increasing redundancy).

Figure 4.11 shows the effect of varying the number of stages on the system anonymity under information leak attacks with a fixed $r = 3$. We can see that for small values of f, increasing l increases anonymity because it makes passive information leak attacks harder to launch. However, there is a crossover point at about $f = 0.17$, where increasing redundancy does not help. This is because of the cascading effect of active attacks: at every successive stage, the probability that nodes in that stage are compromised increases.

Figure 5.3 shows the effect of varying the number of stages on the system anonymity under selective DoS attack, for $r = 3$. We can see that for small values of $\bar{t}$, the benefit of increasing the number of stages to counter passive attacks is very small, because an increase in the number of stages gives more opportunities to the attackers to launch selective DoS attack. Also, the crossover point where increasing l is counterproductive has fallen to about $f = 0.12$. For $f \geq 0.12$, increasing l reduces anonymity, because both conventional active attacks and selective DoS attack become stronger. We conclude that $r = 3$ and $l = 2$ or 3 are optimal design choices.

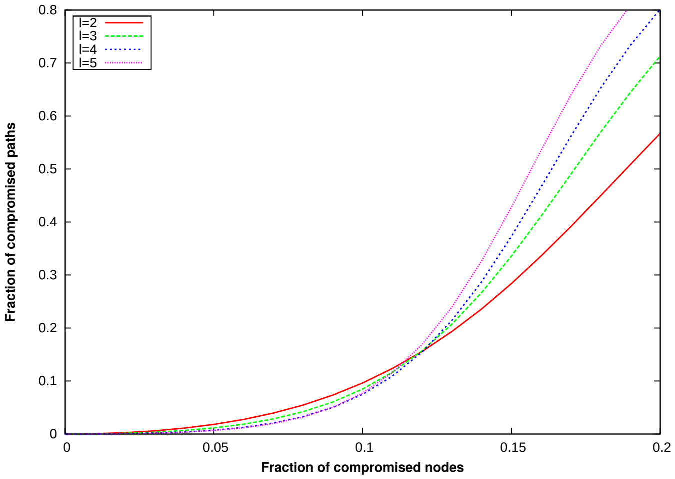


Figure 5.3: Effect of varying l.

5.3 Selective DoS in Lookups

The attack algorithm described above considers the ability of adversarial nodes to deny service when they are part of the path construction mechanism. The adversary could also launch a selective DoS attack on the lookup mechanism. In fact, in our analysis, we have already considered one form of selective DoS on lookup: when all of the redundant lookups are compromised, and there is no malicious node within bounds, the lookup is aborted (see Figure 3.1).

Recently, Tran et al. [21] introduced a new selective DoS attack on the lookup. To resolve conflicting results from redundant lookups, the initiator chooses the closest node that satisfies the bounds check. Tran et al. [21] observed that if the target node is honest, and the adversary is not able to compromise all redundant lookups, then it can simply return an invalid node with an identifier very close to the target identifier. This can be done by precomputing the hashes of all 2^{32} IP addresses, and returning the appropriate IP address. This attack greatly increases the abilty of malicious nodes to deny service; only a single lookup path needs to be compromised to carry out this attack.

We propose a simple defense to this attack: to resolve conflicting results from redundant lookups, the initiator can choose the closest IP address *that speaks the Salsa protocol.* Using this defense, the probability of first stage nodes being honest does not change from our lookup security analysis in Chapter 3. However, for nodes in second and subsequent stages, the initiator cannot directly verify if an IP address is part of the protocol, and has to rely on intermediate nodes to perform this stage. Since some of the malicious intermediate nodes may lie during the verification, the initiator may again get conflicting results. For this scenario, we investigate an *abort on conflict* policy.

We note that in the protocol analyzed by Tran et al. [21] as well as in our proposed defense, each node in the first stage must contact all nodes in the second stage, increasing the threat of information leaks. We will incorporate this into our analysis as well. From Figure 5.4, we can see that the user anonymity is even more reduced as compared to the previous case.

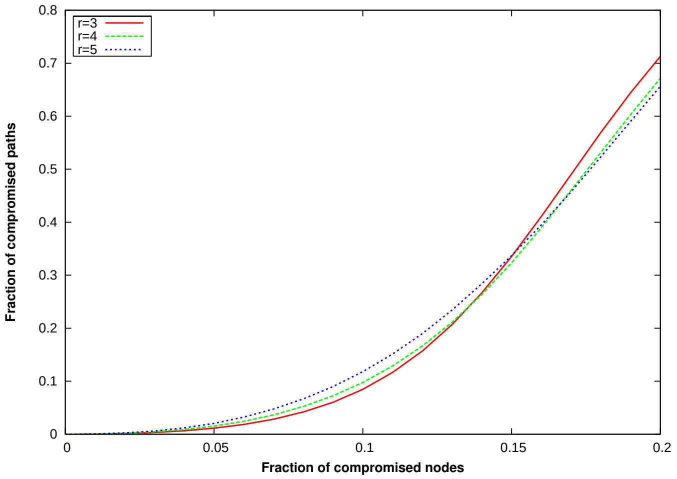


Figure 5.4: Effect of varying r.

CHAPTER 6

AN ENTROPY BASED APPROACH FOR EVALUATION

6.1 Lookups

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, and 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 a complete 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 entropy:

$$H(I) = -\sum_{i} p_i \log_2 i \tag{6.1}$$

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).

$$H(I|O) = \sum_{o \in O} P(o) H(I|o) \tag{6.2}$$

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.

6.1.1 Single Lookup

Clearly, when the lookup is not intercepted by the adversary (null observation), the attacker does not learn any information and the entropy is $\log H$, where H denotes the set of honest nodes. 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 $prev = p$ in the lookup path is $Y = y$ *levels* away from it in the binary tree structure. We have that

$$H(I|O) = \sum_{y,p} P(o = y, p) H(I|o = y, p) \tag{6.3}$$

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

$$P(o = y, p) = \sum_{i \in H} P(o = y, p|I = i) \cdot P(I = i) \tag{6.4}$$

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

$$P(I = i) = \frac{1}{|H|} \tag{6.5}$$

We shall now compute $P(o = y, p|I = i)$. Let us denote the distance between node i and the target in terms of binary tree levels as $D = d_i$. Observe that given $I = i$, there cannot be a jump (in terms of binary tree levels) in the lookup path of size greater than d_i, relative to the next hop. Thus we have that when $d_i < y$, then $P(o = y, p|I = i) = 0$. In the case when $y = 0$, $P(o = y, p|I = i)$ 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 $Y = y$ 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 $P(o = y, p|I = i)$ 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 $\frac{G}{N}$. 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 $\frac{G}{N} \cdot f \frac{2^{y-1}}{G}$ 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 $L = l$ nodes so far (not including the final malicious hop), then we require that these l nodes are 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 $Y = y$, 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 $\binom{d-y-1}{l-2} \cdot (\frac{1}{2})^{d-y-1}$.

Combining the above, we have that

$$P(o = y, p | I = i) =$$
$$\begin{cases} f & y = 0, i = p \\ \frac{G}{N} \cdot f \cdot \frac{2^{y-1}}{G} & i, p \in \text{same group} \\ \sum_{l=2}^{d-y+1} (1-f)^l \cdot f \cdot \frac{1}{2} \frac{1}{2^{y-1}} \cdot \frac{G}{N(1-f)} \cdot \binom{d-y-1}{l-2} \cdot (\frac{1}{2})^{d-y-1} \cdot \frac{2^{y-1}}{G} & \text{otherwise} \end{cases} \tag{6.6}$$

Using $P(I = i)$ and $P(o = y, p | I = i)$ from Equations (6.5) and (6.6), we can now compute $P(o = y, p)$ from Equation (6.4).

Let us now compute $H(I | o = y, p)$. By definition, we have that

$$H(I | o = y, p) = -\sum_{i \in H} P(I = i | o = y, p) \log P(I = i | o = y, p) \tag{6.7}$$

Since we have already computed $P(o = y, p | I = i)$, $P(I = i)$ and $P(o = y, p)$ in Equations (6.6), (6.5) and (6.4), we can use Bayesian inference to compute $P(I = i | o = y, p)$ as follows:

$$P(I = i | o = y, p) = \frac{P(o = y, p | I = i) \cdot P(I = i)}{P(o = y, p)} \tag{6.8}$$

By using $P(o = y, p)$ from (6.4) and $H(I | o = y, p)$ from (6.7), we can compute the entropy of the lookup from Equation (6.3).

6.1.2 Redundant Lookups

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

$$H(I|O) = \sum_{y_1,p_1} .. \sum_{y_r,p_r} P(o_1 = y_1, p_1 .. o_r = y_r, p_r) H(I|o_1 = y_1, p_1 .. o_r = y_r, p_r) \tag{6.9}$$

Similar to the case of single lookup, we can condition the probability $P(o_1 = y_1, p_1 .. o_r = y_r, p_r)$ on the event $I = i$. Using the observation that the redundant lookups are independent given $I = i$ we can compute $P(o_1 = y_1, p_1 .. o_r = y_r, p_r)$ as follows:

$$P(o_1 = y_1, p_1 .. o_r = y_r, p_r) = \sum_{i \in H} P(o_1 = y_1, p_1 .. o_r = y_r, p_r | I = i) \cdot P(I = i) \tag{6.10a}$$

$$P(o_1 = y_1, p_1 .. o_r = y_r, p_r) = \sum_{i \in H} \prod_{j=1}^{r} P(o_j = y_j, p_j | I = i) \cdot P(I = i) \tag{6.10b}$$

where $P(o = y, p|I = i)$ and $P(I = i)$ are given by equations (6.6) and (6.5). Let us now compute $H(I|o_1 = y_1, p_1 .. o_r = y_r, p_r)$.

$$\begin{aligned} &H(I|o_1 = y_1, p_1 .. o_r = y_r, p_r) = \\ &- \sum_{i \in H} P(I = i|o_1 = y_1, p_1 .. o_r = y_r, p_r) \log P(I = i|o_1 = y_1, p_1 .. o_r = y_r, p_r) \end{aligned} \tag{6.11}$$

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

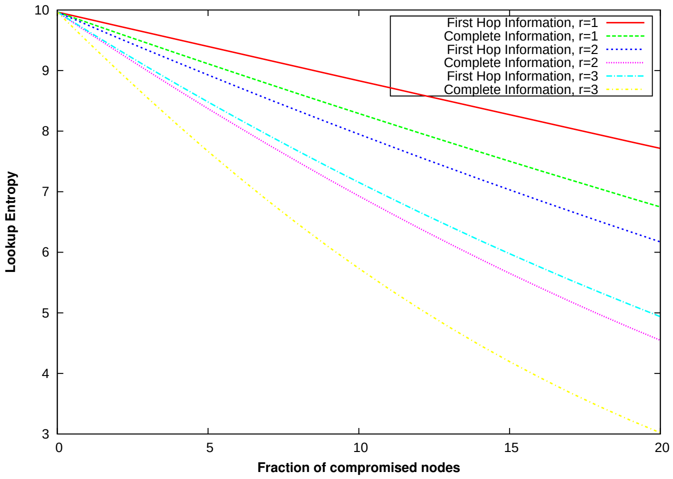


Figure 6.1: Lookup entropy.

$$P(I = i|o_1 = y_1, p_1..o_r = y_r, p_r) = \frac{P(o_1 = y_1, p_1..o_r = y_r, p_r|I = i) \cdot P(I = i)}{P(o_1 = y_1, p_1..o_r = y_r, p_r)} \tag{6.12a}$$

$$P(I = i|o_1 = y_1, p_1..o_r = y_r, p_r) = \frac{\prod_{j=1}^{r} P(o_{=}y_j, p_j|I = i) \cdot P(I = i)}{P(o_1 = y_1, p_1..o_r = y_r, p_r)} \tag{6.12b}$$

Finally, we can use Equation (6.9) to compute the entropy of redundant lookups. Figure 6.1 plots the entropy of the lookup as a function of the fraction of compromised nodes in the system. The input parameters for our model were $N = 1000, g = 128$. We can see that by considering 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 hop. For instance, when the fraction of compromised nodes is $f = 0.2$, incorporating all possible information leaks reduces the entropy from approximately 5 to 3 for a redundancy parameter of $r = 3$. This 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.

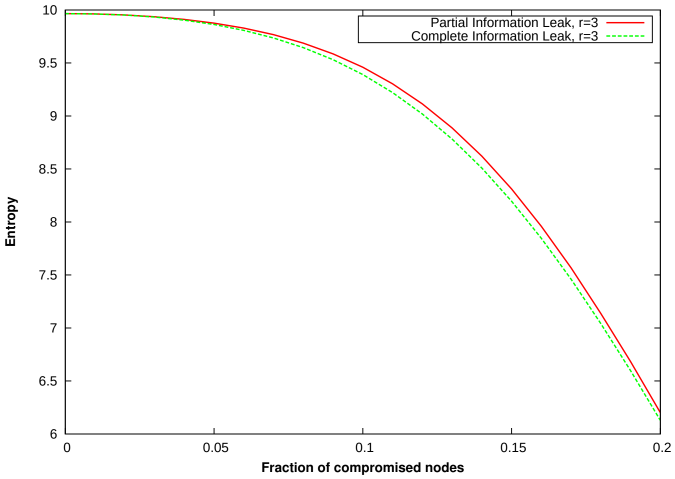


Figure 6.2: Circuit entropy.

6.2 Path Construction

Our entropy based analysis of lookups suggests that the anonymity provided by the path construction mechansim is likely to be even lower than our results shown in Chapter 4. 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 intiator 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 6.2 shows the reduction in the anonymity 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 the future work.

CHAPTER 7

RELATED WORK

Secure routing in peer-to-peer networks has been the subject of a lot of research [22, 23, 8, 5, 24]. We studied lookup mechanisms proposed by Castro et al. [8] and Nambiar and Wright [5], focusing on the information leak from lookups, and observed a trade-off between security and anonymity of a lookup. Kapadia and Triandopoulos recently proposed Halo [24], which is also based on redundant routing, and exhibits a similar trade-off. 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 [25] proposed an anonymous DHT based on Koorde [26], which performs a randomized routing phase before an actual lookup. Ciaccio [27] 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.

Danezis and Clayton [28] studied attacks on peer discovery and route setup in anonymous peer-to-peer networks such as Tarzan [6]. They proposed a node knowledge profiling attack and showed that unless a node learns about a vast majority of the network, the attackers would be able to link it to its traffic with high probability. Note that this attack assumed a global passive adversary, escaping the issue of detecting lookups. We have shown that even a partial adversary can make observations about a large fraction of lookups that occur in the P2P network. Recently, Bauer et al. [29] 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 [30] proposed the predecessor attack, which was later extended by Wright et al. [31, 32, 33]. 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 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 attack 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 probability that both the first and last nodes are attackers also increase resilience against predecessor attacks.

Similar to predecessor attacks, there is a thread of research that deals with degradation of anonymity over a period of time. Berthold et al. [34] and Raymond [13] propose intersection attacks that aim to compromise sender anonymity by intersecting sets of users that were active at the time the intercepted message was sent, over multiple communication rounds. Similarly, Kesdogan et al. [35] use intersection to find recipients of a given users message. A statistical version of this attack was proposed by Danezis [36] and later extended by Mathewson and Dingledine [37]. These attacks typically require an adversary to observe a significant fraction of the network. Information leaks in peer-to-peer systems, however, 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 paper is that, when building anonymous systems, it is important not to abstract away the properties of the system that can affect anonymity. 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 [38, 39], clock skew [40], heterogeneous path latency [41, 39], the ability to monitor Internet exchanges [42] and reliability [20]. Due to lack of space, we only briefly discuss the last two attacks. Conventional anonymity models of Tor view a connection from a client to a server as point to point link, and abstract away the fact that this connection passes through the internet routers. Murdoch and Zieliński [42] showed that Internet exchange-level adversaries were capable of observing a vast majority of this traffic, and could degrade user anonymity by performing end-to-end timing analysis. Borisov et al. [20] proposed a selective-DoS attack on anonymous communication, and showed that attackers could selectively affect the reliability of the system in states that are hardest to compromise. Selective-DoS attack affects peer-to-peer anonymous communication the most, because of the added complexity of knowing only a subset of the nodes in the network.

CHAPTER 8

CONCLUSIONS

Peer-to-peer approaches to anonymous communication have the potential to eliminate the scalability concerns and central vulnerability points of current anonymity systems like Tor. A key challenge in peer-to-peer systems is the ability to locate relays for anonymous communication.

The secure lookup mechanism in Salsa uses redundant routing, which enables 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 analyzed the security of Salsa, under both active and passive attacks. Salsa incorporates redundancy into the path building mechanism, to counter the lookup bias introduced by active adversaries. This makes salsa vulnerable to several passive attacks, including our bridging attacks based on information leaks from lookups. We have demonstrated the tension that exists between while defending against both active and passive adversaries. Defending against active adversaries requires increasing redundancy, which increases the threat of passive attacks. Salsa was previously reported to tolerate upto 20% compromised nodes, but our results show, 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 is fundamental in the sense it even exists in Salsa with a PKI. Also, increasing path lengths to counter our passive attacks only has a limited benefit, and in some cases, it even reduces anonymity.

We also showed that Salsa is vulnerable to a selective DoS attack, where an adversary denies service to a user only when it is unable to break user anonymity. Selective DoS has a devastating effect on the security of Salsa; an adversary with 20% compromised nodes can compromise more than 70% of the paths.

Finally, we perform a first step towards an entropy based evaluation of

Salsa; instead of considering information leaks when only the first step in the lookup mechanism is compromised, we analyze all possible sources of information leaks. We leave a complete analysis of Salsa's path building mechansim using the entropy metric as future work. Our results demonstrate that information leaks are an important part of anonymity analysis of a system and that new advances in the state of the art of P2P anonymous communication are needed.

APPENDIX A

AP3

AP3 [43] is an anonymous communication system built on top of Pastry [44]. The essence of AP3 operation is similar to Crowds [45], 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 who 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 [45]; 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 $H_{k+} = H_k \vee H_{k+1} \vee H_{k+2} \vee ...$ and let I denote the event that attackers identified the initiator correctly. 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

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

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

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.

A.1 Castro et al.'s Secure Lookup

It has been recognized that unprotected DHTs are extremely vulnerable to attacks on the lookup mechanism. First of all, malicious nodes can perform a Sybil attack [14] 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, 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.

Castro et al.[8] designed a suite of mechanisms to counter these attacks. We discuss their mechanisms in context of Pastry [44], a structured peer-to-peer overlay network, though they are applicable to other DHTs. They proposed:

- *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 node ID 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 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. consider the problem of securely routing to the entire replica set, for which a neighbor anycast mechanism is also used. We refer the reader to [8] for a detailed explanation of the techniques.

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 node ID 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 $1-(1-f)^{L+\log_{2^b} N}$, which is quite high for the typical values of $L=16$ and $b=4$. In Figure 4.2, we plot the probability of detection of the lookup initiator as a function of the fraction of compromised nodes f. We can see that a small fraction of 5% 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.

A.2 The E_1 Attack

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 $P(I|E_1)$:

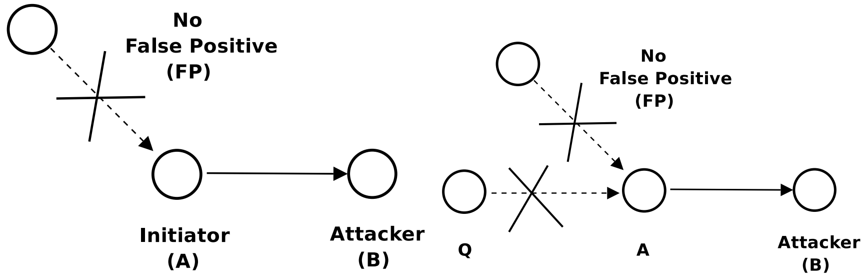
$$P(I|E_1) = \frac{P(I \wedge E_1)}{P(E_1)}$$

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 A.1 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 exclusive, and FP and Q are independent from H_1, H_{2+}, and each other, we can write



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

Figure A.1: Information leak in AP3.

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

Therefore,

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

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 also that as $P(Q)$ grows smaller, the fraction approaches closer to 1. As we noted in Chapter 4, with the secure lookup due to Castro et al., $P(Q)$ is quite small, even for small f.

Figure A.2 shows the attacker confidence as a function of the fraction of the nodes that are compromised for varying p, using $N = 1000$, $b = 4$ and $L = 16$. 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 $p \rightarrow 1$, 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. suggested mechanisms solve this problem, but introduced another, as the lookup is no longer anonymous and can be observed by malicious nodes.

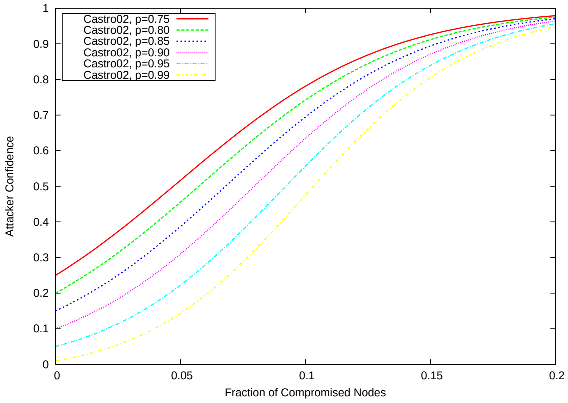


Figure A.2: $P(I|E_1)$.

A.3 The E_i Attack

In addition to E_1, the 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 when 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 following equation:

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

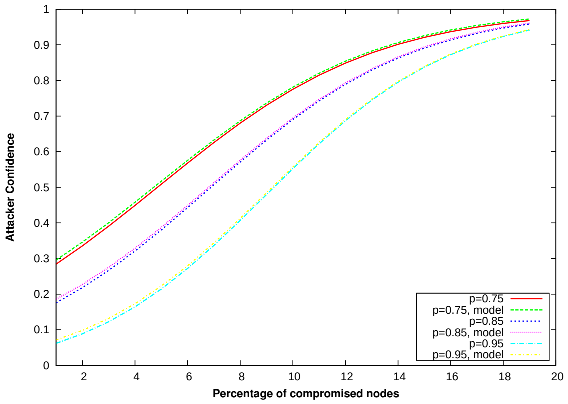


Figure A.3: $P(I|E_2)$.

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 $P(I|E_2 \wedge H_{3+}) = 0$.

$$P(I|E_2) = \frac{P(I|E_2 \wedge H_2)P(E_2 \wedge H_2)}{P(E_2 \wedge H_1) + P(E_2 \wedge H_2) + P(E_2 \wedge H_{3+})} \qquad \text{(A.3)}$$

Figure A.3 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

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

The above formulation neglects false positives and is only an approximation. However, in practice, the approximation works quite well. In Figure A.3, 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 probabilities of the attackers observing

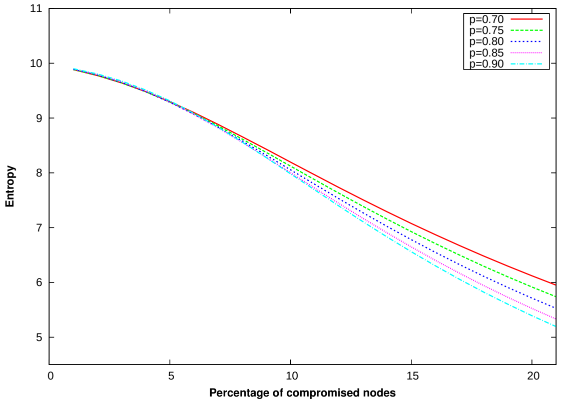


Figure A.4: Entropy as a function of f.

E_1 and E_2. We use the entropy metric of anonymity [46, 47] 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 $\frac{1-P(I|E_i)}{N-1}$.[1] Let $H(E_i)$ be the entropy of the system under the observation E_i. Then, the average entropy can be computed as follows:

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

Figure A.4 plots the entropy as a function of f, for varying p, using $N = 1000$. Note that higher values of p have *lower* entropy, and are thus considered to provide worse anonymity under the entropy metric. This is because 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?

[1] 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.

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The second-generation onion router,” in *Proceedings of the 13th USENIX Security Symposium*, August 2004, pp. 21–21.

[2] P. Palfrader, “Number of running Tor routers, April 2008,” http://www.noreply.org/tor-running-routers/.

[3] D. Goodin, “Tor at heart of embassy passwords leak,” *The Register*, September 10, 2007.

[4] A. Mislove, G. Oberoi, A. Post, C. Reis, P. Druschel, and D. S. Wallach, “AP3: Cooperative, decentralized anonymous communication,” in *Proceedings of the ACM SIGOPS European Workshop*, 2004, p. 30.

[5] A. Nambiar and M. Wright, “Salsa: A structured approach to large-scale anonymity,” in *CCS ’06: Proceedings of the 13th ACM conference on Computer and communications security.* New York, NY, USA: ACM, 2006, pp. 17–26.

[6] M. J. Freedman and R. Morris, “Tarzan: A peer-to-peer anonymizing network layer,” in *9th ACM Conference on Computer and Communications Security*, Washington, DC, November 2002, pp. 193–206.

[7] M. Rennhard and B. Plattner, “Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection,” in *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2002)*, Washington, DC, USA, November 2002, pp. 91–102.

[8] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, “Secure routing for structured peer-to-peer overlay networks,” in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, December 2002, pp. 299–314.

[9] G. Danezis, R. Dingledine, and N. Mathewson, “Mixminion: Design of a Type III Anonymous Remailer Protocol,” in *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003, pp. 2–15.

[10] U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman, "Mixmaster Protocol — Version 2," IETF Internet Draft, July 2003.

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

[12] P. Tabriz and N. Borisov, "Breaking the collusion detection mechanism of MorphMix," in *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006)*, G. Danezis and P. Golle, Eds. Cambridge, UK: Springer, June 2006, pp. 368–384.

[13] J.-F. Raymond, "Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems," in *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, H. Federrath, Ed. Springer-Verlag, LNCS 2009, July 2000, pp. 10–29.

[14] J. Douceur, "The Sybil Attack," in *Proceedings of the 1st International Peer-To-Peer Systems Workshop*, March 2002.

[15] D. Dagon, C. Zou, and W. Lee, "Modeling botnet propagation using time zones," in *In Proceedings of the 13 th Network and Distributed System Security Symposium NDSS*, 2006.

[16] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling, "Measurements and mitigation of peer-to-peer-based botnets: a case study on storm worm," in *LEET'08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*. Berkeley, CA, USA: USENIX Association, 2008, pp. 1–9.

[17] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17–32, 2003.

[18] D. Daly, D. D. Deavours, J. M. Doyle, P. G. Webster, and W. H. Sanders, "Möbius: An extensible tool for performance and dependability modeling," in *Computer Performance Evaluation: Modelling Techniques and Tools*, B. R. Haverkort, H. C. Bohnenkamp, and C. U. Smith, Eds., vol. 1786. Schaumburg, IL: Springer, Mar. 2000, pp. 332–336.

[19] A. Nambiar and M. Wright, The Salsa Simulator, http://ranger.uta.edu/~mwright/code/salsa-sims.zip.

[20] N. Borisov, G. Danezis, P. Mittal, and P. Tabriz, “Denial of service or denial of security? How attacks on reliability can compromise anonymity,” in *Proceedings of CCS 2007*, October 2007, pp. 92–102.

[21] A. Tran, N. Hopper, and Y. Kim, “Hashing it out in public: common failure modes of dht-based anonymity schemes,” in *WPES ’09: Proceedings of the 8th ACM workshop on Privacy in the electronic society*. New York, NY, USA: ACM, 2009, pp. 71–80.

[22] E. Sit and R. Morris, “Security considerations for peer-to-peer distributed hash tables,” in *IPTPS ’01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*. London, UK: Springer-Verlag, 2002, pp. 261–269.

[23] D. Wallach, “A survey of peer-to-peer security issues,” in *International Symposium on Software Security*, Tokyo, Japan, 2002, pp. 42–57.

[24] A. Kapadia and N. Triandopoulos, “Halo: High-assurance locate for distributed hash tables,” in *Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS)*, February 2008, pp. 61–79.

[25] N. Borisov, “Anonymous routing in structured peer-to-peer overlays,” Ph.D. dissertation, Berkeley, CA, USA, 2005, chair-Eric A. Brewer.

[26] M. F. Kaashoek and D. R. Karger, “Koorde: A simple degree-optimal distributed hash table,” in *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS ’03)*, 2003, pp. 36–43.

[27] G. Ciaccio, “Improving sender anonymity in a structured overlay with imprecise routing,” in *Proceedings of the Sixth Workshop on Privacy Enhancing Technologies (PET 2006)*, G. Danezis and P. Golle, Eds. Cambridge, UK: Springer, June 2006, pp. 190–207.

[28] G. Danezis and R. Clayton, “Route Fingerprinting in Anonymous Communications,” *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing*, pp. 69–72, 2006.

[29] K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, “Low-resource routing attacks against Tor,” in *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2007)*, Washington, DC, USA, October 2007, pp. 11–20.

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

[31] M. Wright, M. Adler, B. N. Levine, and C. Shields, "An analysis of the degradation of anonymous protocols," in *Proceedings of the Network and Distributed Security Symposium.* IEEE, February 2002.

[32] M. Wright, M. Adler, B. N. Levine, and C. Shields, "Defending anonymous communication against passive logging attacks," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003, p. 28.

[33] M. Wright, M. Adler, B. N. Levine, and C. Shields, "The predecessor attack: An analysis of a threat to anonymous communications systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 4, no. 7, pp. 489–522, November 2004.

[34] O. Berthold, H. Federrath, and M. Köhntopp, "Project anonymity and unobservability in the Internet," in *CFP '00: Proceedings of the Tenth conference on Computers, Freedom and Privacy.* New York, NY, USA: ACM, 2000, pp. 57–65.

[35] D. Kesdogan, D. Agrawal, and S. Penz, "Limits of anonymity in open environments," in *Proceedings of Information Hiding Workshop (IH 2002)*, F. Petitcolas, Ed. Springer-Verlag, LNCS 2578, October 2002, pp. 53–69.

[36] G. Danezis, "Statistical disclosure attacks: Traffic confirmation in open environments," in *Proceedings of Security and Privacy in the Age of Uncertainty, (SEC2003)*, Gritzalis, Vimercati, Samarati, and Katsikas, Eds., IFIP TC11. Athens: Kluwer, May 2003, pp. 421–426.

[37] N. Mathewson and R. Dingledine, "Practical traffic analysis: Extending and resisting statistical disclosure," in *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, ser. LNCS, vol. 3424, May 2004, pp. 17–34.

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

[39] A. Back, U. Möller, and A. Stiglic, "Traffic analysis attacks and trade-offs in anonymity providing systems," in *Proceedings of the Information Hiding Workshop (IH 2001)*, I. S. Moskowitz, Ed. Springer-Verlag, LNCS 2137, April 2001, pp. 245–257.

[40] S. J. Murdoch, "Hot or not: Revealing hidden services by their clock skew," in *Proceedings of CCS 2006*, October 2006, pp. 27–36.

[41] N. Hopper, E. Y. Vasserman, and E. Chan-Tin, "How much anonymity does network latency leak?" in *Proceedings of CCS 2007*, October 2007, pp. 1–28.

[42] S. J. Murdoch and P. Zieliński, "Sampled traffic analysis by Internet-exchange-level adversaries," in *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*, N. Borisov and P. Golle, Eds. Ottawa, Canada: Springer, June 2007, pp. 167–183.

[43] A. Mislove, G. Oberoi, A. Post, C. Reis, P. Druschel, and D. S. Wallach, "AP3: Cooperative, decentralized anonymous communication," in *ACM SIGOPS European Workshop*, M. Castro, Ed. New York, NY, USA: ACM, 2004, p. 30.

[44] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Middleware*, November 2001, pp. 329–350.

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

[46] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Privacy Enhancing Technologies Workshop*, ser. Lecture Notes in Computer Science, R. Dingledine and P. Syverson, Eds., vol. 2482. Berlin / Heidelberg / New York: Springer, April 2002, pp. 184–188.

[47] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Privacy Enhancing Technologies Workshop*, ser. Lecture Notes in Computer Science, R. Dingledine and P. Syverson, Eds., vol. 2482. Berlin / Heidelberg / New York: Springer, April 2002, pp. 259–263.

[48] R. Dingledine and P. Syverson, Eds., *Workshop on Privacy Enhancing Technologies, April 14–15, 2002, San Francisco, CA, USA*, ser. Lecture Notes in Computer Science, vol. 2482. Berlin / Heidelberg / New York: Springer, April 2002.

AUTHOR'S BIOGRAPHY

Prateek Mittal was born in New Delhi, India. He received a B.TECH in Computer Science and Engineering from the Indian Institute of Technology at Guwahati in 2006. From 2006 onwards, he has been a research assistant at the Hatswitch Security Research group, under the supervision of Prof. Nikita Borisov at the University of Illinois at Urbana-Champaign. At various times. he has also been employed with INRIA at Rocquencourt, France, Microsoft Research at Cambridge, UK, and International Computer Science Institute at Berkeley, USA. His research interests are in the domain of computer networks and distributed systems, focusing on network security, privacy enhancing technologies like anonymous communications and traffic analysis, and robust peer-to-peer networks.