IMPROVING PRIVACY AND PERFORMANCE IN ANONYMOUS COMMUNICATIONS

by

NAYANTARA MALLESH

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2010

Copyright © by NAYANTARA MALLESH 2010

All Rights Reserved

To Daddy and Mom

ACKNOWLEDGEMENTS

I thank my advisor Dr. Matthew Wright for giving me the opportunity to be his student and for his guidance during the course of my Ph.D. I thank him most for his patience and his belief in my ability to do better.

I also thank my committee members Dr. Chengkai Li, Dr. Donggang Liu, and Dr. Gergely Zaruba for being part of my thesis committee and for giving their insight and constructive criticism of my work. Their enthusiasm has led me to work harder and strive for better ideas.

I would like to express my gratitude to the Department of Computer Science and Engineering for giving me the opportunity to work as a teaching assistant. These years with the department have given me invaluable experience that I hope to use when I become a teacher. I give my fondest gratitude to UTA, my alma mater, for being my home during the trials and tribulations of my grad school years.

Most of all I thank my parents, Wg. Cdr. Mallesh and Mrs. Gopika Mallesh for always believing in me. I could not have come this far without your unconditional love and support. Thank you Rishi and Praneeta for cheering me on during the ups and the downs. Thank you Rohan, for being my best friend, my supporter through it all, and for always being there. You are my rock.

I thank Rakesh for his friendship, company, coupons, and the many discussions we have had in GACB, 252, 244, and enroute to our favorite eat-outs in DFW.

November 19, 2010

ABSTRACT

IMPROVING PRIVACY AND PERFORMANCE
IN ANONYMOUS COMMUNICATIONS

NAYANTARA MALLESH, Ph.D.
The University of Texas at Arlington, 2010

Supervising Professor: Matthew Wright

Anonymous communications systems provide an important privacy service by keeping passive eavesdroppers from linking communicating parties. However, an attacker can use long-term statistical analysis of traffic sent to and from such a system to link senders with their receivers. While it is important to protect anonymous systems against such attacks, it is also important to ensure they provide good performance. In this thesis, we aim to make contributions to both these areas.

In the statistical disclosure attack (SDA), an eavesdropper isolates his attack against a single user, whom we call Alice, with the aim of exposing her set of contacts. To study the SDA we introduce an analytical method to bound the time for the eavesdropper to identify a contact of Alice, with high probability. We analyze the attack in different scenarios beginning with a basic scenario in which Alice has a single contact. Defenses against this attack include sending cover traffic, which consists of sending dummy messages along with real messages. We extend our analysis to study the effect of two different types of cover traffic on the time for the attack to

succeed. We further extend our analysis to investigate the effectiveness of the attack for a partial eavesdropper who can observe only a part of the network. We validate our analysis through simulations and show that the simulation results closely follow the results of analysis. Although our bounds are loose, they provide a way to compare between different amounts and types of cover traffic in various scenarios.

In the second part of this thesis, we investigate how cover traffic can be used as an effective counter strategy against the SDA. We propose that the mix generate cover traffic that mimics the sending patterns of users in the system. This *receiver-bound cover (RBC)* helps to make up for users that arent there, confusing the eavesdropper. We show through simulation how this makes it difficult for the eavesdropper to discern cover from real traffic and perform attacks based on statistical analysis. Our results show that receiver-bound cover substantially increases the time required for this attack to succeed. When our approach is used in combination with user-generated cover traffic, the attack takes a very long time to succeed.

The original statistical disclosure attack has focused on finding the receivers to whom Alice sends. In this part of the thesis, we investigate the effectiveness of statistical disclosure in finding all of Alices contacts, including those from whom she receives messages. To this end, we propose a new attack called the *Reverse Statistical Disclosure Attack (RSDA)*. RSDA uses observations of all users sending patterns to estimate both the targeted users sending pattern and her receiving pattern. The estimated patterns are combined to find a set of the targeted users most likely contacts. We study the performance of RSDA in simulation using different mix network configurations and also study the effectiveness of cover traffic as a countermeasure. Our results show that that RSDA outperforms the traditional SDA in finding the users

contacts, particularly as the amounts of user traffic and cover traffic rise.

In the final part of this thesis, we study how a sparse network topology affects the security of anonymous systems. We show that an expander topology such as a sparse, D-regular graph exhibits security properties comparable to a fully connected graph; in a reasonable number of hops and even for small values of degree D. Further, we show that if the expander graph is constructed with a bias towards lower round-trip time links, there is a considerable gain in performance without compromise in security.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Privacy is a basic need. In the past few years, there has been a phenomenal increase in the use of the Internet in our daily lives. The amount of personal data collected and stored electronically, has exploded. Devices with increasing capabilities and multiple funtionalities are making access to the Internet possible anytime, from anywhere.

Activies like personal communication, information gathering, payment of bills, travel reservations, and obtaining directions, is regularly being done on the Internet. Such usage of the Internet generates a trail of information about individual users. Personal information like age, home location, work location, educational background, friends, and business contacts can be gathered from an individual's online activities. This information is not always in the form of disconnected pieces of data. Sewn together it can mirror the actually state of people's physical lives. Worse, not all of this private information is visible only to the parties that need the data in order to provide a service.

Anonymous communications systems allow users to achieve some level of privacy in the online world. In this thesis, we study the security provided by anonymous communciations systems. We study a particular type of attack on high latency anonymity systems called the statistical disclosure attack. Furthermore, we mathematically model this attack in order to better understand and analyze how it performs under different conditions. We then go on to study possible defenses against this type of attack, in particular, using cover traffic.

We also propose an enhancement over the traditional attack called the Reverse Statistical Disclosure Attack (RSDA) that uses more of the available information to make the attack much faster. We study how much defenses against SDA can be used against RSDA. We show that the enhanced RSDA attack is not slowed significantly by using cover traffic alone as a defense.

In the final part of the thesis, we study how the topology of the network affects the security and performance of the anonymous communications system. We study the impact of restricting the topology of the network from a fully connected topology to a restricted topology in which each node is connected to few other nodes. In a restricted topology each node has a much smaller degree, which is the number of neighbors it connects to, compared to a fully connected topology in which the degree of each node is is $(N-1)$ for an N node network. In particular, we look at constructing the network based on an expander topology, such as a sparse D-regular graph, and study how the topology affects the security and performance properties of the anonymous system.

CHAPTER 2

BACKGROUND

2.1 Anonymous Communications Systems

Anonymous communications systems keep eavesdroppers from linking communicating parties.

2.1.1 Low Latency Anonymous Systems

Low latency anonymous systems can be used to relay traffic for applications like web browing and instant messaging. Such applications are sensitive to response time and the anonymous system must provide minimum performance requirements.

2.1.1.1 Tor

The onion router *(TOR)* is a popular low latency anonymous system. It consists of an overlay network of volunteer nodes that relay traffic to and from users and their destinations.

2.1.2 High Latency Anonymous Systems

High latency anonymous systems are useful for applications that can tolerate some amount of delay in transmission. Such applications include email, and publishing to blogs and bulletin board services.

2.2 The Disclosure Attack

The family of Disclosure Attacks are a type of long-term intersection attacks that target a single user with the aim of exposing the contacts of that user. The original Disclosure Attack [1] proposed by Kesdogan et. al. uses a simple threshold mix [2] to deanonymize Alice's contacts. In each round that Alice participates, the attacker records the set of receivers who receive messages. The attacker makes observations until he is able to obtain m mutually disjoint receiver sets ensuring that each set contains a single recipient of Alice. This part of the attack is called the *learning phase*. The attacker then refines a receiver set by intersecting it with a new observation only if the observation also intersects with none of the other $m-1$ sets. It turns out that the learning phase is an NP-complete problem which makes this attack extremely difficult to carry out in practice. The Statistical Disclosure Attack [3] describes a more practical way of carrying out the Disclosure Attack.

2.3 The Statistical Disclosure Attack

SDA is a probability-based approach to the disclosure attack and is a practical way to link a single targeted user Alice to her set of contacts [3]. The SDA is based on observations of the number of messages entering the mix network and the distribution of receivers receiving those messages in each round. A *round* is the period of time during which a mix collects messages, mixes and then flushes them out to receivers. The eavesdropper makes observations in a number of *rounds* in which Alice sends messages. In each round of observation i, the eavesdropper records three pieces of information: n_i, the number of messages sent by Alice; m_i the number of messages sent by senders other than Alice; and $\overrightarrow{o_i}$ the distribution of messages received by receivers in that round. Senders other than Alice are known as *background* senders.

In order to learn background senders' behavior, the attacker makes observations in multiple rounds in which Alice does not participate. The eavesdropper records the distribution of messages from background senders to receivers in a vector $\overrightarrow{u_i}$ in each round of observation i. The eavesdropper averages $\overrightarrow{u_i}$ values over a large number of observations to obtain $\overline{U}$ which represents the sending behavior of background senders. The attacker sums $\overrightarrow{o}$ values over a large number of observations to obtain $\overline{O}$. Since $\overrightarrow{o}$ is recorded when both Alice and background senders participate, it represents their combined sending behavior. Since $\overline{O}$ represents the combined sending behavior of both Alice and the background during the observed rounds, and this can be written as:

$$\overrightarrow{O} = nD_A + m\overrightarrow{U} \tag{2.1}$$

Here n and m are the total number of messages sent by Alice and the background, respectively, during the attacker's observation period. D_A is a vector that represents Alice sending behavior. Ideally, $0 < D_A[r] < 1$ if receiver r is a contact of Alice and $D_A[s] = 0$ if receiver s is not Alice's contact. $\overrightarrow{U}$ is obtained by averaging observations u_i in rounds which Alice does not participage i.e. $\overrightarrow{U} = \sum_{i=1}^{T} \overrightarrow{u_i}$. where T is the number of rounds of observation. If the attacker is unable to collect background statistics before Alice begins communicating, $\overrightarrow{U}$ can be approximated as $\overrightarrow{U}[r] = \frac{1}{R}\ \forall$ r, meaning that the background sends in a uniform manner to all receivers. Alice's most likely set of contacts are determined by solving for D_A in equation (2.1) and picking c receivers with the highest $D_A[r]$ values.

Here n_A is the total number of messages sent by Alice in during the period of observation. Alice's likely set of recipients can be determined by solving equation (2.1) for D_A. The indices with the highest values in D_A correspond to the most likely recipients of Alice.

2.4 PMDA

The Perfect Matching Disclosure Attack [4] (PMDA) is another attack in the family of Disclosure Attacks. PMDA models each round as a bipartite graph. The set of senders and set of receivers each forms one of the two partitions of the bipartite graph while messages from senders to receivers represent the links connecting the two graph partitions. The use of this model allows each round to be represented as a linear assignment problem, which is solved to maximize the joint probability of the links from senders to receivers. The PMDA is a two-step attack. In the first step, called the *profiling step*, the attacker profiles each sender using traditional SDA. This step yields a vector per sender wherein each element contains the probability of the corresponding receiver being a contact of that sender. In the second step, the attacker uses the probabilties obtained in the earlier step to find the maximum weighted bipartite matching on the graph. The maximum bipartite matching is easily found by solving the associated linear assignment problem. The PMDA approach is different from SDA and TS-SDA because in those attacks Alice is the target user and all users apart from Alice are observed as a group and not individually. The PMDA is especially relevant because both RSDA and PMDA use traditional SDA to profile multiple users as compared to a single user. In the PMDA, all *senders*are profiled using traditional SDA. Borrowing terminology from PMDA, we can say that in the RSDA, all *receivers* are profiled using traditional SDA. In the Perfect Matching Disclosure Attack (PMDA) [5], the attacker attempts to improve on SDA by using the insight that only one sender could have sent a particular message. This is best explained by a simple example in the threshold mix setting. Suppose that Alice and Bob are senders and Carol and Dave are receivers. In a given round, suppose that Alice and Bob each send one message and Carol and Dave each receive one message. Based on prior observations (profiling using SDA), both Alice and Bob are

more likely to have sent to Carol than Dave. Since only one of them sent to Carol, however, PMDA finds the most likely matching of senders to receviers with, say, Alice sending to Carol and Bob sending to Dave. This matching is used to inform the profile of each sender and improve the attacker's chances of finding Alice's contacts.

This use of other senders' profiles is used in an entirely different way from RSDA. In particular, Alice is never a receiver and messages received by senders are never used in the profiling. We believe that the traffic analysis improvement in PMDA is therefore largely orthogonal to RSDA. Since both techniques require profiling of the users, however, combining the insights of PMDA with those of RSDA is challenging and we leave this for future work.

2.5 TS-SDA

The Two-sided Statistical Disclosure Attack - When Alice sends a message, she may be *initiating* the message or she may be replying to a message initiated by another user. If the attacker is only interested in knowing to whom Alice initiates messages, the SDA may have problems, as it is not designed to distinguish replies from initiated messages. The Two-sided Statistical Disclosure Attack (TS-SDA) [6] extends the original SDA with observations of messages sent to Alice. TS-SDA uses these additional observations to estimate the likelihood that a given message from Alice is a reply to a previously received message and discounts possible replies accordingly.

TS-SDA is based on very different assumptions from the RSDA. In particular, the assumption that the attacker is only interested in receivers of Alice's initiated messages leads TS-SDA to filter out the statistical influence of possible replies. In the current work, we assume that the attacker is interested in all of Alice's contacts, whether Alice initates the communication or not. TS-SDA would thus be worse than SDA in our model.

A message S_i from Alice could be either an initiated message or a reply to a message from an earlier round. The likelihood that a message is initiated by Alice, Z_I, can be approximated using Alice's rate of initiation, the number of messages she has sent so far and number of rounds observed. The probability that the message is a reply, Z_R, can be calculated based on Alice's reply probability and reply delay with respect to messages received by her in previous rounds. If S_i is a reply to a previous message R_j, then the distribution $\overrightarrow{I_{ij}} = \frac{\overrightarrow{o_i}.\overrightarrow{s_i}}{|\overrightarrow{o_i}.\overrightarrow{s_i}|}$ describes the likely receiver of S_i. $\overrightarrow{I_{ij}}$ is an intersection of the receivers, $\overrightarrow{o_i}$ who received messages in the round S_i was sent and $\overrightarrow{s_j}$, the set of receivers who sent messages in the round R_j was sent. So in a round that Alice sends a message, S_i,

$$\overrightarrow{o} = n_A.\frac{Z_I.D_A + Z_R.\sum_j \overrightarrow{I_{ij}}}{Z_I + Z_R} + n_B.D_N \tag{2.2}$$

Here j is the number of messages that Alice *received* in prior rounds. D_A is calulated from the above equation in each round and the mean over multiple rounds, $\widehat{D}_A$ is obtained. A more detailed description of this attack is provided in [6]. Simulation results in [6] show that three quarters of the time, TS-SDA found the correct recipient of Alice's message in the top 20 candidate receivers whereas SDA placed the correct receiver in the top 35. The results also indicate that as the time between receiving a message and replying to the message increased, both SDA and TS-SDA became less effective. TS-SDA was found to be better than SDA at uncovering recipients of Alice's replies. However, both attacks were equally effective in uncovering recipients of messages initiated by Alice to her contacts.

2.6 Hitting Set Attack

The Hitting Set Attack [7] offers polynomial (SHS-attack) and superpolynomial (HS*-attack) time work-arounds to the NP-complete learning phase of the Disclosure

Attack. Instead of checking if all possible receiver sets fulfill the hitting set property, these work-arounds identify good candidate sets and then check only if the good candidates are acceptable solutions.

2.7 Peer-to-peer Anonymous Systems

System for anonymous peer-to-peer services, such as GNUnet [8], Freenet [9], and APFS [10], include receivers in the system by their nature. Sending cover traffic to receivers would be very reasonable in such systems. P5 is an anonymity system that provides sender, receiver, and sender-receiver anonymity[11]. P5 creates a hierarchy of broadcast channels with each level providing a different level of tradeoff between anonymity and communication performance. In P5, noise (dummy) messages are added to prevent statistical correlation of sources and sinks of a communication stream. Real messages and noise messages move from the source to the sink hop by hop across different nodes. Intermediate nodes cannot distinguish real packets from dummy packets and treat all transiting packets similarly. Furthermore, intermediate nodes are also sources and insert dummy packets into outgoing streams. Dummies are dropped at the final destination. By using these channels, each sender effectively creates a form of receiver-bound cover traffic, as each message is sent to a group of receivers. While this multicast approach would be one way to do receiver-bound cover traffic in mix-based anonymity systems, it would only work in non-encrypted communications.

CHAPTER 3

AN ANALYSIS OF SDA

In this chapter, we present an analysis of the performance of SDA and bound the number of rounds for SDA to be successful. We compare the bounds for different scenarios in order to understand how different design choices impact the performance of SDA. We first introduce the network model we used for the analysis and then discuss the assumptions we in this analysis. We then give an overview of our approach and finally validate the analysis using simulation results.

3.1 Model

We assume that there are N senders that wish to communicate with a set of R recipients using a mix network. We will generally set $R = N$ for simplicity, but the relationship between senders and receivers is many-to-many. In each *round* of communication, a set of senders send messages via the mix network to a set of receivers. The mix network may consist of a single mix or a network of connected mixes. For simplicity, we abstract away the details of the number of mixes in the mix network and refer to a single mix or a cascade of mixes as a *mix*.

3.1.1 Mix Types

For our analysis we consider is a simple threshold mix, which collects a batch of B messages in each round and forwards them in a random order to their destinations.

3.1.2 Alice and Background Senders

Alice is one of the senders and is the target of the eavesdropper. Senders other than Alice are called *background senders*.

3.1.3 Communication Links and Sending Behavior

Background senders send m messages per round to receivers. These messages are distributed uniformly and randomly among the R receivers. The average number of messages a single receiver gets in any round is $\frac{m}{R}$. Receivers who do not receive from Alice are called non-Alice receivers.

Alice has a single recipient called Bob. In each round, Bob receives one message from Alice and $\frac{m}{R}$ messages on average from other senders. Thus, Bob receives on average $1 + \frac{m}{R}$ messages in rounds that Alice sends and $\frac{m}{R}$ messages in rounds that Alice does not send.

3.1.4 Attacker Model

The attacker is a global passive adversary who can observe all links from senders to the mix and all links from the mix to recipients. The target of the attacker is Alice and the attacker's aim is to expose the set of recipients with whom Alice communicates. The attacker observes multiple rounds , including rounds with and without Alice's participation, and tries to identify Alice's recipients. The attacker can observe only the incoming and outgoing links from the mix and cannot observe activity inside the mix network. This assumption is for the simplicity of the model, as there are many configurations for a mix network, but also because the statistical disclosure attack is effective without observations of activity inside the network. The attacker makes observations over a number of rounds T.

3.1.5 Partial Attacker

For an attacker to be able to see all communication going into and out of the anonymity system, he must be very powerful. Such adversaries are not impossible but are rare. One way a partial attacker can be implemented is as described in [12]. By gaining control of a number of internet exchanges (IX), the attacker is able to see a part of the traffic going into and out of a number of autonomous systems (AS) zones. Anonymity systems such as Mixminion and Tor contain nodes in many different AS zones. The adversary who has control of a number of IXes would be able to observe all of the traffic going in and out of these IXs.

3.1.6 Cover Traffic

Cover traffic consists of dummy messages that are inserted into the network along with real user messages. Dummy messages have long been recognized as a useful tool to increase anonymity provided by mix-based systems. For the purpose of analysis, we consider two types of cover traffic based on where it is generated. *Alice cover* is cover traffic generated by Alice herself . On the other hand, *receiver-bound cover (RBC)* is generated by the mix and sent to message recipients.

3.2 Approach

The analysis proceeds by considering the number of messages that each receiver gets in each round of observation. After a sufficient number of rounds of observation T, the attacker will observe that Bob receives more messages than a typical receiver in rounds that Alice sends. We find an upper bound on T such that the attacker observes a minimum difference between typical receivers and Bob.

We derive equations to calculate bounds on T in different scenarios, beginning with the case when there is no cover traffic. We extend the analysis to bound T from

above in the presence of Alice cover and receiver-bound cover. Finally, we derive bounds for the number of rounds of observation needed by a partial adversary who observes only a fraction of the network.

3.3 Analysis

In this section we present our analysis of the statistical disclosure attack. We begin our analysis with a basic scenario in which Alice has a single contact Bob and uses no cover traffic. We then extend the analysis to include cover traffic from Alice. In the next part of the analysis we study the impact of receiver-bound cover traffic on the performance of the attack. Finally, we consider the case when a the attacker is a partial eavesdropper who can observe only a part of the network.

3.3.1 Bounding the time for attacker success

Let $X = \sum_{i=1}^{T} X_i$ be the number of messages Bob receives after T rounds of observation. $X_i = 1 + \frac{m}{R}$ is the total number of messages sent to Bob from both Alice and the background senders in round i. Let the expected value of X after T rounds, $\mu_x = T\left(1 + \frac{m}{R}\right)$.

Let $Y = \sum_{i=1}^{T} Y_i$ be the number of messages received by each receiver other than Bob, after T rounds of observation. $Y_i = \frac{m}{R}$ is the total number of messages from background senders to a typical receiver in round i. Let the expected value of Y after T rounds of observation is $\mu_y = T\left(\frac{m}{R}\right)$.

We use a Chernoff bound to find the lower bound T_{low}, of the number of rounds for Bob to receive less than a threshold B messages with a probability $p_L = 0.5$. We choose a probability of 0.5, which means that 50% of the time Bob will have more than B messages because it maps to the median rounds metric used in our simulations and allows for a comparison between the analysis and simulation results. For the threshold

B we choose the mid-point between μ_x and μ_y the expected messages/round for Bob and for a typical non-Alice receiver, respectively. The mid-point is an arbitrary boundary chosen for the value of B because it provides, in the long run, a place between the X and Y distributions.

$$B = T\left(\frac{m}{R} + \frac{1}{2}\right) \tag{3.1}$$

Using the Chernoff bounds [13] for $0 < \delta_x \leq 1$ we get,

$$Pr[X < (1-\delta_x)\mu_x] < exp\left(\frac{\mu_x(\delta_x)^2}{2}\right) \tag{3.2}$$

We want the probability that $X < B$ to be $p_L = 0.5$. So,

$$exp\left(\frac{\mu_x(\delta_x)^2}{2}\right) = \frac{1}{2} \tag{3.3}$$

and using B as the lower bound

$$(1-\delta_x)\mu_x = B$$

From (3.1), $B = \frac{Tm}{R} + \frac{T}{2}$

$$(1-\delta_x)\mu_x = T\left(\frac{m}{R} + \frac{1}{2}\right)$$

$$(1-\delta_x)T\left(1 + \frac{m}{R}\right) = T\left(\frac{m}{R} + \frac{1}{2}\right)$$

$$\delta_x = \frac{R}{2(m+R)} \tag{3.4}$$

Substituting (3.4) in (3.3),

$$exp\left(\frac{-T_{low}(1+\frac{m}{R})\left(\frac{R}{2(m+R)}\right)^2}{2}\right) = \frac{1}{2}$$

$$T_{low} = 8ln(2)\left(1 + \frac{m}{R}\right) \tag{3.5}$$

We now use a Chernoff bound to derive an upper bound T_{up} on the number of rounds of observation required for a 0.5 or lower probabability that a receiver other than Bob will get more than B messages. As before, we choose a probability of 0.5 because it maps to the median rounds metric used to measure attacker success in our simulations and allows for a comparison between the analysis and simulation results.

Let p_U be the probability that a non-Alice receiver receives more than B messages. Then, the probability that at least one of the $R-1$ non-Alice receivers gets more than B messages is $p = 1-(1-p_U)^{R-1}$. We want the probability p to be lower than 0.5 as mentioned earlier. If we choose $p_U = \frac{1}{R}$ then for $R = 1000$, for example, $p = 1-(1-0.001)^{999} = 0.63$ which is higher than the desired probability of 0.5 chosen earlier. However, if we choose $p = \frac{1}{2R}$, then $p = 1-(1-0.002)^{999} = 0.39$ which is lower than the desired probability of 0.5. Hence, we choose $p = \frac{1}{2R}$ for deriving the upper bound T_{up}.

Applying the Chernoff bound with $\delta_x > 0$,

$$Pr[Y > (1+\delta_y)\mu_y] < \left(\frac{e_y^{\delta}}{(1+\delta_y)^{1+\delta_y}}\right)^{\mu_y} \tag{3.6}$$

We want $Pr[Y > B]$ to be $\frac{1}{2R}$. So,

$$Pr[Y > (1+\delta_y)\mu_y] < \frac{1}{2R} \tag{3.7}$$

Using B as the upper bound

$$(1+\delta_y)\mu_y = B$$

$$(1+\delta_y)\mu_y = T\left(\frac{m}{R}+\frac{1}{2}\right)$$

$$\delta_y = \frac{R}{2m} \tag{3.8}$$

Substituting (3.8) in (3.7) and taking the log of 2 both sides, we get

$$\frac{mT_{up}}{R} ln\left(\frac{e^{\delta_y}}{(1+\delta_y)^{1+\delta_y}}\right) = ln\left(\frac{1}{2R}\right)$$

$$T_{up} = \frac{R}{m} ln\left(\frac{1}{2R}\right)\frac{1}{\delta_y - (1+\delta_y)ln(1+\delta_y)} \tag{3.9}$$

3.3.2 Bounding the time for attacker success in the presence of Alice cover

Let us now assume that Alice decides to use cover traffic to increase her privacy. Let p_d be the probability that a message from Alice is a dummy message. The probability that a message from Alice is a real message is $r = 1 - p_d$. Thus, the expected number of messages that Bob receives after T rounds is $\mu_x = T\left(\frac{m}{R} + r\right)$. We find the lower bound for μ_x in the same way we did before:

$$Pr[X < (1-\delta_x)\mu_x] < \frac{1}{2}$$

$$exp\left(\frac{-\mu_x\delta_x^2}{2}\right) = \frac{1}{2} \tag{3.10}$$

We set $(1-\delta_x)\mu_x = B$,

$$(1-\delta_x)\left(\frac{m}{R}+r\right) = \left(\frac{m}{R}+\frac{r}{2}\right)$$

$$\delta_x = \frac{Rr}{2(m+Rr)}$$

Substituting for δ_x in (3.10) we get

$$exp\left(\frac{-T_{low}\left(\frac{m}{R}+r\right)\left(\frac{Rr}{2(m+Rr)}\right)^2}{2}\right)=\frac{1}{2}$$

$$T_{low}=8ln(2)\frac{m+Rr}{Rr^2} \tag{3.11}$$

where $r=(1-p_d)$

To get the upper bound on the number of rounds of observation we proceed as before and obtain $\delta_y=\frac{Rr}{2m}$ and

$$T_{up}=\frac{R}{m}ln\left(\frac{1}{2R}\right)\frac{1}{\delta_y-(1+\delta_y)ln(1+\delta_y)} \tag{3.12}$$

3.3.3 Bounding the time for attacker success in the presence of RBC

In the presence of receiver-bound cover traffic, Bob receives $\frac{mV_{rbc}}{R}$ more messages than before. V_{rbc} is the volume of receiver-bound cover. So if the outgoing traffic in a round is $\frac{m}{R}$, the amount of RBC sent in that round is $V_{rbc}\frac{m}{R}$. This means $\mu_x = T\left(\frac{m}{R}(1+V_{rbc})+r\right)$. Following the derivation steps as before, we get $\delta_x=\frac{Rr}{2(m+mV_{rbc}+Rr)}$ and

$$T_{low}=8ln(2)\frac{m+Rr+mV_{rbc}}{Rr^2} \tag{3.13}$$

In the presence of RBC, other recievers also receive $\frac{mV_{rbc}}{R}$ more messages than before. This means $\mu_y=T.\frac{m}{R}(1+V_{rbc})$. Using Chernoff bounds to bound the number of rounds of observation, we get $\delta_y=\frac{Rr}{2m(1+V_{rbc})}$ and

$$T_{up}=ln\left(\frac{1}{2R}\right)\frac{R}{m(1+V_{rbc})}\frac{1}{\delta_y-(1+\delta_y)ln(1+\delta_y)} \tag{3.14}$$

3.3.4 Bounding the time for success of a partial attacker

In this subsection we use Chernoff bounds to derive an upper limit on the number of rounds for a partial attacker to successfully identify Bob as a contact of Alice. Since the partial adversary can observe only parts of the network, we assume that a message entering or exiting the anonymity network has a probability p_p of being observed by the attacker. p_p varies from 0 to 1 depending on how much of the network the attacker can observe; 0 means he can observe nothing and 1 means the attacker is a global adversary and is able to observe every message going in and coming out of the mix network.

We proceed similarly to Section 3.3.1 and include the observation capability of the attacker, p_p, into our derivation steps. The number of messages the adversary sees Bob receiving after T rounds is now $X = \sum_{i=1}^{T} X_i p_p$. The number of messages the adversary observes other receivers receiving is $Y = \sum_{i=1}^{T} Y_i p_p$. The expected values of X and Y after T rounds of observation is $E[X] = \mu_x = T\left(\frac{m}{R} + 1\right) p_p$ and $E[Y] = \mu_y = T\left(\frac{m}{R}\right) p_p$. We set B to be half the distance between μ_x and μ_y, which means $B = T p_p \left(\frac{m}{R} + \frac{1}{2}\right)$. Using Chernoff bounds and proceeding as before we get $\delta_x = \frac{R}{2(m+R)}$ which is the same as (3.4) and

$$T_{low} = \frac{8ln(2)}{p_p}\left(1 + \frac{m}{R}\right) \tag{3.15}$$

$\delta_y = \frac{R}{2m}$ which is the same as (3.8) and

$$T_{up} = \frac{R}{mp_p} ln\left(\frac{1}{2R}\right) \frac{1}{\delta_y - (1+\delta_y) ln(1+\delta_y)} \tag{3.16}$$

From the above bounds we note that for a partial adversary, the number of rounds required to achieve the same certainty as a full attacker increases by a factor of p_p.

3.3.5 Combined Equations

Combining equations (3.5), (3.11), (3.13), and (3.15) we get

$$T_{low} = \frac{8ln(2)}{p_p} \frac{m + Rr + mV_{rbc}}{Rr^2} \tag{3.17}$$

and combining equations (3.9), (3.12), (3.14), and (3.16) we get

$$T_{up} = \frac{R}{mp_pV_{rbc}} ln\left(\frac{1}{2R}\right) \frac{1}{\delta_y - (1+\delta_y)ln(1+\delta_y)} \tag{3.18}$$

3.4 Comparison of Analysis and Simulation Results

We now compare the bounds given by the analysis in Section 3 with simulations of SDA using a threshold mix. For the simulations we set the simulation parameters to match the assumptions made in the analysis so that the analysis and simulations results are comparable. We set the number of senders and receivers to be $N = R = 1000$. We assume Alice has one contact called Bob. Alice participates in some rounds and does not particpate in other rounds. She sends one message/round to Bob in rounds that she participates. The probability that Alice is online and sends messages to Bob is determined by p_{online}. For the comparison simulations we set $p_{online} - 1.0$. We assume that the eavesdropper has already observed rounds in which Alice does not participate and has an understanding of background senders' behavior. The mix batch size is set to $B = 125$ messages/round. In the simulations we measure the median rounds for attacker success. This means, in Alice rounds, 50% of the time Bob is the highest receiver compared with other receivers. In the analysis, we set $p_L = 0.5$ which means that 50% of the time, Bob receives more than B messages and non-Alice receivers get less than B messages. This makes the analysis and simulation results comparable. Though the analysis provides a somewhat loose bound on the number

of rounds of observation, it more importantly makes possible a way to understand SDA under different scenarios and compare the effectiveness of different types and quantities of cover traffic as a counter-measure.

We now compare and discuss the analysis and simulation results that are presented it Figures 3.1 to 3.6. In our first set of results shown in Figure 3.1 and Figure 3.2 we compare the performance of SDA when no cover traffic is present, when Alice cover traffic is present, and when varying amounts of receiver-bound cover traffic is introduced. The simulations show that when no cover traffic is used, the attack takes less than 40 rounds to succeed even when there are large amounts of background messages to mix with Alice's messages. The analysis shows that Alice cover provides some benefit to Alice. The simulations results validate the analysis as seen in Figure 3.2. In the presence of Alice cover, the number of rounds for the attack increases to 190. When receiver-bound cover is introduced, the analysis predicts an approximately 2-fold, 3-fold, and 4-fold increase in the time for attacker success compared to when only Alice cover is used. The simulation results shown in Figure 3.2 agree with the analysis and show, for example, when $m = 2000$ messages/round of background traffic and $RBC = 100\%$ the number of rounds jumps to 400. When $RBC = 200\%$ is introduced, the number of rounds almost triples to 560 and when $RBC = 300\%$ is used along with Alice cover, the number of rounds close to quadruples and it takes 720 rounds for the eavesdropper to succeed.

Figures 3.3 and 3.4 show the performance of SDA, in analysis and simulation respectively, for varying amounts of Alice cover and receiver-bound cover. The number of Alice dummies per round is sampled from a geometric distribution whose success probability p_d is varied from 0.1 to 0.9. The number of dummies from Alice per round increases from 0.1 to 9 dummy messages/round during the simulation and this is plotted along the x-axis. The analysis shows that as Alice cover increases, the attack's

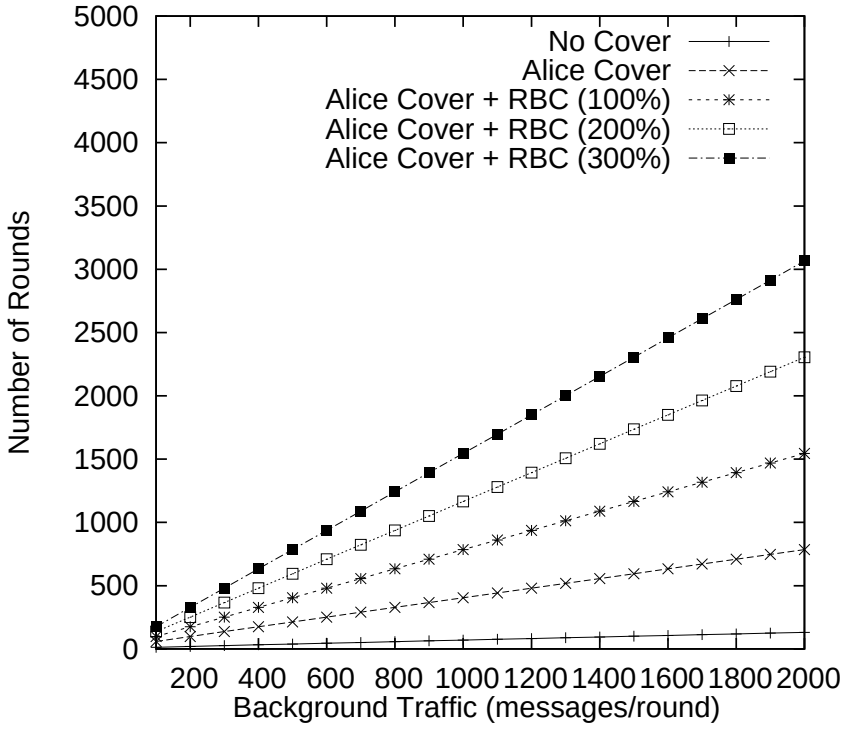

Figure 3.1. **Analysis:** Median rounds for Bob to receive more messages than any other user with increasing background traffic.

time to succeed increases more than 65 times when Alice cover alone is used and more than 75 times when RB cover is used along with Alice cover. The simulation results validate the analysis and show an increase in attack time of 50 and 60 times for the same scenarios.

Next, we show analysis and simulation results for a partial eavesdropper who can see only half of the links into and out of the mix, i.e for $p_o = 0.5$. These results are shown in Figures 3.5 and 3.6 for the analysis and simulation respectively. The time taken for SDA to succeed when different amounts of cover traffic is used shows an increase similar to the global eavesdropper results discussed earlier. Additionally, the analysis shows that when the attacker is a partial eavesdropper who can observe only half the network, the time taken for the attack to succeed doubles. This is exactly

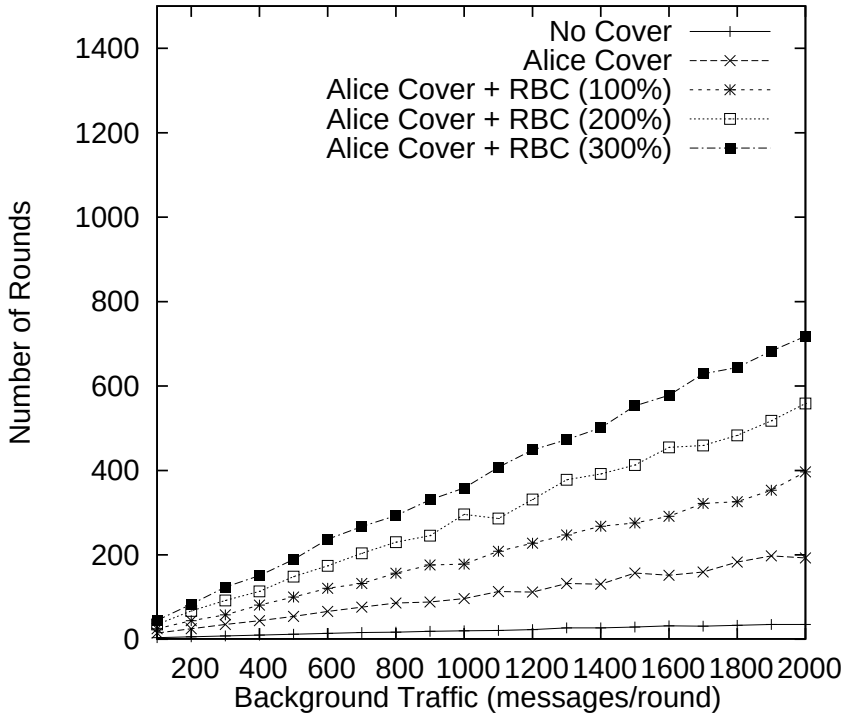

Figure 3.2. **Simulation:** Median rounds to find Bob with increasing background traffic. Threshold mix, batch size = 100 to 500 messages/round.

what the simulation results show; comparing the simulation results partial and full eavesdropper we see that the number of rounds of observation in Figure 3.6 show a 75% to 100% increase over the observation rounds in Figure 3.4.

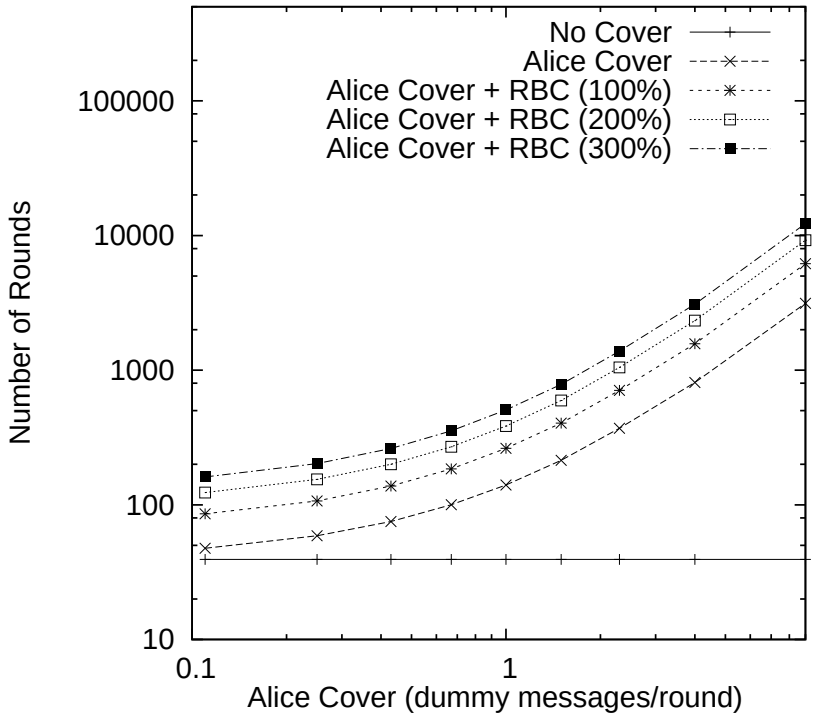

Figure 3.3. **Analysis:** Median rounds for Bob to receive more messages than any other user with increasing cover traffic from Alice.

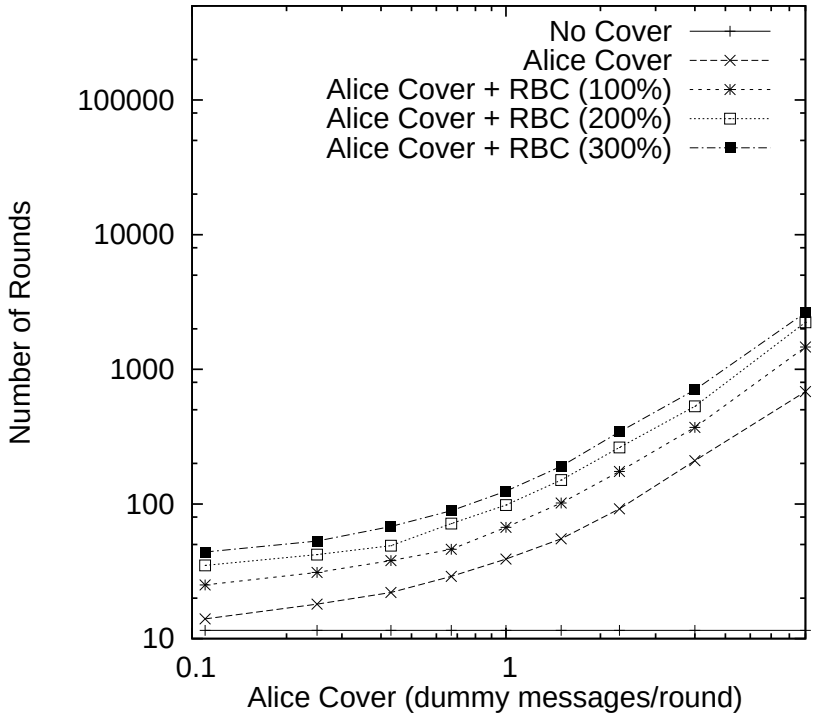

Figure 3.4. **Simulation:** Median rounds to find Bob with increasing cover traffic from Alice. Threshold mix with batch size = 500 messages/round.

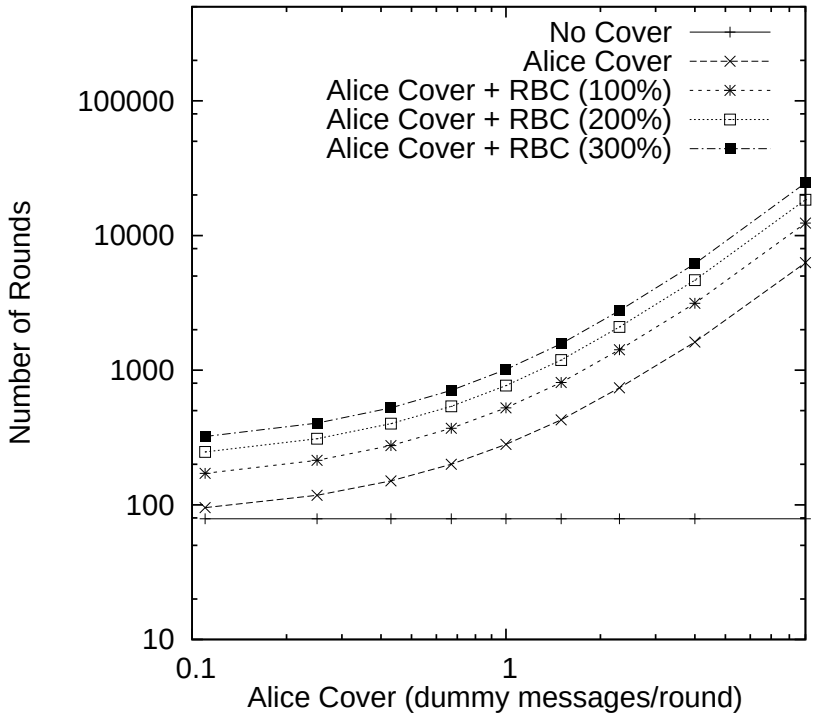

Figure 3.5. **Analysis:** Median rounds for Bob to receive more messages than any other user with increasing Alice cover for a partial observer.

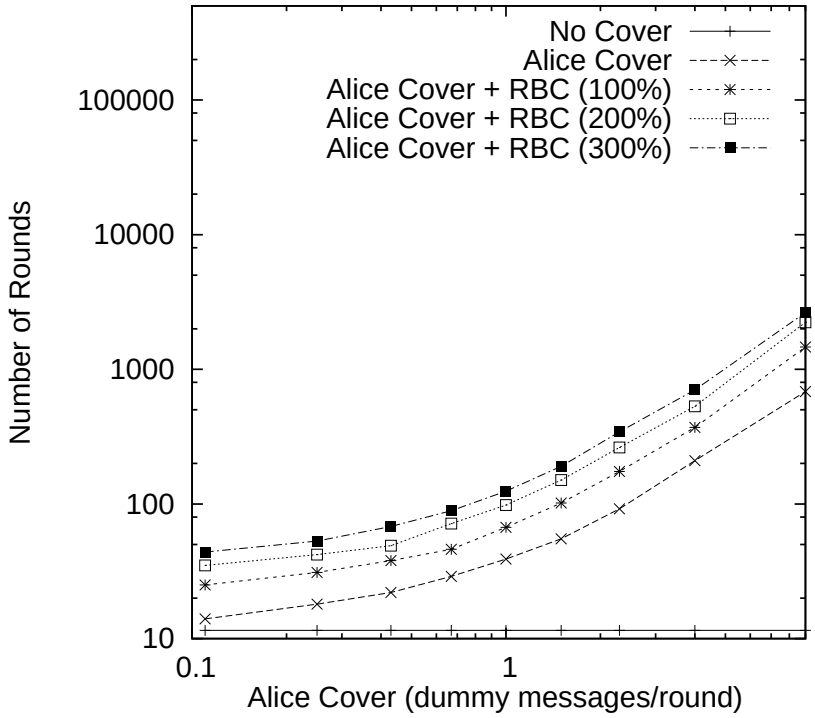

Figure 3.6. **Simulation:** Median rounds to find Bob with increasing Alice cover for a partial observer. Threshold mix, batch = 500 mesg/round.

CHAPTER 4

RECEIVER-BOUND COVER TRAFFIC

4.1 Introduction

Anonymity systems are fundamentally challenging to build on top of the existing Internet architecture. The simplest and most secure approaches require all participants to send messages at the same rates, e.g. one message per given time interval. Users without a message to send must send fake messages, known as *cover traffic* or *dummies*, to ensure anonymity for themselves as well as for others. This provides no allowance for the realities of node failure and network partitions. Furthermore, users are not online all the time and at the cost of their own benefit are unable to provide consistent cover traffic.

Existing implementations based on the mixes paradigm introduced by Chaum [2] remove this unrealistic requirement for constant participation, but at a cost to their security. The changing group of users can be observed, along with outgoing messages, leading to powerful *intersection attacks*. In these attacks, differences in the membership of the set of users are matched with the differences in the message-sending behavior, leading to links between users and their receivers. Effectively, the attacker can observe information leaks over time.

In this chapter, we first explain the network model used, then we discuss why cover traffic delays statistical disclosure and how it can be used to counter this attack.

4.2 Model

Mathewson and Dingledine developed a simulation, including the use of a binomial mix (called a pool mix in their paper), to investigate the effect of a number of parameters on the performance of SDA [14]. They found that as the number of Alice's contacts grew, the rounds of observation to expose her full contact list correspondingly increased. They also found that cover traffic from Alice was effective in slowing, but not preventing, SDA. Cover traffic from Alice was found to be more effective when the delay probability of the binomial mix was increased. Increasing the mix delay spreads out the incoming traffic over a number of outgoing rounds, making it more difficult for the attacker to estimate which set of receivers might have gotten the messages from Alice.

4.2.1 Cover Traffic

We are not the first to propose sending cover traffic to receivers. Berthold et. al have users send pre-generated dummy messages to the recipient when the sender is offline [15]. Mathewson and Dingledine suggest, and then dismiss, this approach in a footnote of their work on statistical disclosure [14]. They cite problems with the user sending to all receivers, which we avoid by having the mix generate the cover traffic. Shmatikov and Wang propose cover traffic sent to receivers to prevent active and passive timing analysis attacks in low-latency mix networks [16]. In their approach, senders generate the dummies in advance and send them to the mix, which later sends them when cover traffic is needed. The authors point out that dummy packets sent on the link between the mix and recipient can be easily recognized and dropped by the recipient. Mix-generated cover traffic is also useful in protecting reverse paths from malicious clients that use the Overlier-Syverson attack. The results from Section 5.5 of our work indicate that this approach can also help prevent intersection attacks.

Mathewson and Dingledine extend SDA to pool mixes [14]. Their work relaxes some of the assumptions made in the original work [3].

4.2.2 Model

Let us assume that there are N senders that wish to communicate with a set of R recipients using a mix network. We will generally set $R = N$ for simplicity, but the relationship between senders and receivers is many-to-many. In each *round* of communication, a set of senders send messages via the mix network to a set of receivers. The mix network may consist of a single mix or a network of connected mixes. For simplicity, we abstract away the details of the number of mixes in the mix network and refer to a single mix or a cascade of mixes as a *mix*.

4.2.3 Mix Types

For our study we consider two types of mixes. The first type of mix is a simple threshold mix, which collects a batch of B messages in each round and forwards them in a random order to their destinations. The second type of mix called a binomial mix [17] applies a weighted coin-toss to each incoming message to decide if the message leaves the mix in the current round or is delayed until a later round. The binomial mix uses a delay probability of P_{delay} to bias the decision.

4.2.4 Communication Links and Sending Behavior

We model the relationships between senders and receivers as a scale-free network, in which the distribution of node degrees follows a power law relationship [18]. This means that most senders communicate with a few well-known recipients in addition to many lesser-known recipients. The well-known recipients hence receive more messages during their communication lifetime. Senders other than Alice are called

background senders. Background senders send more frequently to their well-known recipients than their lesser-known recipients. Alice uses a *uniform method* of sending and sends uniformly to all her recipients.

4.2.5 Attacker Model

The attacker is a global passive adversary who can observe all links from senders to the mix and all links from the mix to recipients. The target of the attacker is a sender Alice and the attacker's aim is to expose the set of recipients with whom Alice communicates. The attacker observes multiple rounds , including rounds with and without Alice's participation, and tries to identify Alice's recipients. The attacker can observe only the incoming and outgoing links from the mix and cannot observe activity inside the mix network. This assumption is for the simplicity of the model, as there are many configurations for a mix network, but also because the statistical disclosure attack is effective without observations of activity inside the network.

4.2.5.1 Partial Attacker

For an attacker to be able to see all communication going into and out of the anonymity system, he must be very powerful. Such adversaries are not impossible but are rare. One way a partial attacker can be implemented is as described in [12]. By gaining control of a number of internet exchanges (IX), the attacker is able to see a part of the traffic going into and out of a number of autonomous systems (AS) zones. Anonymity systems such as Mixminion and Tor contain nodes in many different AS zones. The adversary who has control of a number of IXes would be able to observe all of the traffic going in and out of these IXs.

4.2.6 Cover Traffic

In reality, most senders are not online all the time. It is difficult for users to consistently send cover traffic, as it requires them to be online and connected to the mix network all the time. This problem is potentially alleviated when the mix carries the onus of sending cover traffic. In the rest of this paper, we study the effectiveness of padding generated by the users and by the mix. We discuss cover traffic in more detail in the next section.

4.3 Cover Traffic

Cover traffic consists of dummy messages that are inserted into the network along with real user messages. Dummy messages have long been recognized as a useful tool to increase anonymity provided by mix-based systems. In the context of our model, cover traffic can be classified into three types based on where it is generated. *user cover* is cover traffic generated by Alice herself and *background cover* is cover traffic generated by other senders connecting to the mix. On the other hand, *receiver-bound cover (RB)* is generated by the mix and sent to message recipients.

Mathewson et.al. have shown that user cover helps delay statistical analysis [14]. When Alice generates cover traffic with a geometric distribution, she can significantly delay SDA. A more effective approach is for Alice to send a threshold number of messages in every round. If the number of real messages is less than the threshold, then Alice inserts dummy messages to compensate for the shortage. Both of these approaches become more effective as the mix exhibits higher delay variability, since the number of possibilities that the attacker must consider increases. Even if the sender is online 100% of the time, however, sender-originated dummy packets alone are not enough to protect against statistical analysis.

4.3.1 Background Cover Traffic

Background cover is created when many mix users generate dummies along with their real messages. Cover traffic from users other than Alice could be seen as providing cover for Alice's messages. Note that the users have a strong incentive to provide these dummies, as it helps to protect their own privacy. As we show in Section 5.5, this can be very effective in confusing a naive attacker. However, a slightly more sophisticated attacker can account for background cover and reduce its effectiveness.

We now describe how the naive attacker proceeds in the presence of background cover traffic. The attacker uses the Equation 2.1 to find D_A which contains an estimate of Alice's recipients. In each round, the attacker observes a number of messages entering and exiting the mix. He estimates the number of (i) Alice messages exiting the mix, n_{Alice} and (ii) the number of background messages exiting the mix per round, $n_{Background}$. These estimates are calculated from the mix's delay policy and on the number of messages seen entering the mix from Alice and from the other users. The attacker records the set of recipients who receive messages in each round in $\overrightarrow{r}$, which contains an element for every recipient in the system. $\overrightarrow{r}[i]$ contains the number of messages received by the i^{th} recipient in a particular round. $\overline{O}$ is updated each round as follows:

$$\overline{O}[i] = \frac{\overrightarrow{r}[i] * n_{Alice}}{n_{Alice} + n_{Background}}$$

When background dummies are sent, the attacker sees more messages entering the mix. The dummies get dropped inside the mix and do not exit the mix along with real messages. The attacker, however, expects the messages to exit the mix and wrongly estimates the value of $n_{Background}$. As a result the calculation of $\overline{O}$ is upset, thereby affecting the number of rounds to correctly identify Alice's recipients.

To counter background cover, the attacker can discount away a percentage of incoming messages that he knows are dummies. We assume that the background user's policies for sending dummies are known to the attacker. This can be reasonable in many systems, as only the aggregate behavior is needed. Such policies may be observed by subtracting the number of real output messages from the number of input messages over a period of time in which Alice is not active. We show in Section 5.5 that background dummies do not help against this informed attacker, and that Alice cannot rely on help from her fellow users.

4.3.2 Receiver-bound Cover Traffic

Receiver-bound (RB) cover consists of dummy messages generated by the mix. The dummies are inserted into outgoing user traffic in every round. The mix chooses the recipients of cover traffic uniformly and randomly from the list of recipients. $\overline{O}[i]$ contains the probability that a message received by the i^{th} recipient has originated at Alice. The attacker updates elements in $\overline{O}$ in every round according to Equation 4.3.1. When RB dummies are present, elements in $\overline{O}$ are wrongly updated for messages that were in fact never sent by any sender. This upsets the attackers' statistical calculations. In order for the attack to be successful, the number of rounds the attacker must observe increases significantly. We discuss the practical issues with this approach in Section 4.6.

4.4 Simulation

Using the basic sender-mix-receiver model described in Section 5.2, we simulate the process of sending messages, cover traffic, and the corresponding SDA. We first discuss the three main elements of the simulation design, which are the attacker

algorithm, the generation of real traffic, and our metrics for attacker success. We then describe how we generate cover traffic.

4.4.1 Simulator Design

We built our simulations around the core simulator used by Mathewson and Dingledine, and we refer the reader to that paper for further detail [14].

4.4.1.1 Attacker Algorithm

A full attacker is able to see all messages from senders into the anonymity system and all messages exiting the system to receivers. A partial attacker can see part of the network and can only see some of the messages from senders to the mix system (inbound) and from the mix system to receivers (outbound). To simulate a partial attacker we use a probability $p_p = 0.5$ to decide whether the attacker sees a particular inbound message or outbound message.

The attacker algorithm is based on the statistical analysis approach *Attacking pool mixes and mix networks* described in [14]. Beyond this, we assume that the attacker makes reasonable adjustments to the algorithm in response to changes in the system, such as adjustments to background cover described in Section 4.3.

4.4.1.2 Attacker adjustment to background cover

The attacker can estimate the average total background cover from the set of background sends. In the presence of background cover, the attacker discounts the expected background cover per round from the total number of messages sent by background senders in every round.

4.4.1.3 Attacker adjustment to receiver-bound cover

When discounting RBC the attacker cannot simply discount the number of estimated dummy messages from the total number of messages as he does in the case of background cover. In the case of RBC, the attacker must discount RBC on a per receiver basis in order to preserve the distribution of the number of messages received by each receiver in a round. By proportionally discounting dummy messages from all receivers, the attacker discounts on average the total numberof estimated dummy messages in each round.

In our simulations, we discount receiver-bound cover by applying a discount to each message coming out of the mix. If the mix's RBC volume is V_{rbc}, then the discount is $\frac{1}{1+\frac{V_{rbc}}{100}}$ per message. This means in the presence of RBC, the attacker counts a fraction of the volume of actual traffic each receiver gets in any given round.

4.4.1.4 Real Message Generation

Major elements in the simulated generation of real messages include:

- Background Traffic: To ensure comparability with previous empirical work, the number of messages sent by the background follows a normal distribution with mean 125 and standard deviation of 12.5. Additionally, we consider a more active set of users, with means of 1700 and 9000 messages per round. The senders follow a scale-free model in sending to recipients. We first created a scale-free network and then created a weighted recipient distribution for background senders. The weighted distribution allows background senders to send more messages to popular recipients. A uniform recipient distribution is created for Alice, which allows Alice to send uniformly to all of her recipients.

- Alice's Traffic: Alice has a recipient set of 32 recipients. In each round she sends messages to recipients chosen with uniform probability from this set. Alice generates real messages according to a geometric distribution with a distribution parameter of 0.6, which means that she sends about 1.5 real messages per round.
- Mix Behavior: We use two different mix types for our simulations. For the threshold mix simulations, the batch size is set at 125 messages/round. For simulations to compare the analysis and simulations the batch size is varied from 100 to 2000 messages per round.

  In the case of a binomial mix, the mix applies a probability P_{delay} to each message entering the mix in order to decide if the message will exit the mix in the current round or will be delayed until a later round [17]. For our simulations we varied P_{delay} from 0.1 to 0.9. For simulations where P_{delay} does not vary, we set $P_{delay} = 0.1$.

4.4.1.5 Measuring Attacker Success

For most of our experiments, we measure the number of rounds that the attacker takes to correctly identify ten of Alice's recipients. This is a deviation from prior work, which chose to determine when the attacker correctly identified all 32 of her recipients. The latter is, in our opinion, an unnecessarily high bar for the attacker to meet. In particular, we discovered that finding the final recipient was a particularly challenging task that took many additional rounds of communication in most experiments. Worse, the variance for obtaining this final recipient is quite high, as it may depend on just a few messages that are sent with low probability.

We propose the lower threshold of ten recipients, although arbitrary, as a point at which the attacker has identified a substantial fraction of Alice's recipients. At this point, the attacker can correctly identify not only the popular members of Alice's

recipient set, but also several of the less popular members as well. The attacker may not have the full profile that he seeks, but some of Alice's privacy has been lost, as the attacker has some picture of Alice's communication patterns. Since the attack could take many rounds, a partial picture may be all that the attacker could attain in a reasonable time frame.

It should be noted that we stop all runs after one million rounds. This could equate to almost one hundred and fifteen years, at one hour per round, or nearly two years at one minute per round. If the attacker cannot identify 10 of Alice's recipients in this time, the attack is taking very long. Even if the attacker is that patient, and Alice is that consistent, we focus our attention on stopping the attacker from defeating the system in a faster time frame. When we have strong methods for doing that, longer term attacks can be considered.

4.4.2 Cover Traffic Scenarios

The simulations in [14] focus mainly on the effects of user cover traffic. In this study, we describe the effects of RB cover and background cover. We use three scenarios to evaluate the effect of cover traffic on statistical analysis.

4.4.2.1 Alice and Background Cover Traffic

We first study how dummy messages sent by users other than Alice affects statistical analysis. We set $N = 2^{16}$ as the number of senders. Each of the $N - 1$ other senders apart from Alice, called background senders, generate zero or more dummy messages in every round. Senders choose the number of dummies according to a geometric distribution with a parameter varying from 0.1 to 0.9. This means each sender sends between 0.11 to 9 dummy messages per round on average.

Alice also generates a number of dummy messages in each round in which she participates. Like other senders, Alice follows a geometric distribution to select the number of dummies to send per round. Alice's dummy parameter, P_{dummy}, is varied from 0.1 to 0.9. In simulations where Alice's cover traffic does not vary, we set P_{dummy} to 0.6, which is about 1.5 messages/round. The geometric distribution parameters for Alice dummies and background dummies are independent of each other. Cover traffic generated by senders is sent to the mix like real traffic. The mix can recognize real messages from dummies and drops all dummies that it receives. Hence, dummies sent from the users are dropped inside the mix network and are not propagated to any receivers.

4.4.2.2 Receiver-Bound Cover Traffic

We also evaluate how RB cover traffic originating at the mix impacts SDA. At the end of each round, the mix selects a subset of messages in its pool and sends them to their respective recipients. In addition to the real messages, the mix adds a number of dummy messages to the outbound stream. We run simulations with the number of receiver-bound dummy messages per round, $V_{rbc} = 100\%$, 200%, and 300% of real traffic. The recipient of each dummy message is chosen uniformly at random from the set of recipients. Although the mixes may not know the full set, a reasonable approximation can be constructed by using previously observed recipients and a selection of receiver addresses from the general population. Dummy messages travel from the mix to the receiver and are observed as part of the outgoing traffic by the passive attacker. However, since the attacker cannot distinguish dummy messages from real messages, dummies are included in the attackers analysis. Dummy messages reach the destination nodes and are dropped by the receiver.

4.4.2.3 Alice and Receiver-bound Cover

In this scenario, Alice sends cover traffic to the mix along with her real messages. These messages are dropped inside the mix. The mix in turn generates dummy messages independent of Alice's dummy messages. The mix dummies are sent out with real outbound user messages.

4.5 Results

In this section we present the results of our simulations. Please note the use of logarithmic scales in some of our graphs. The Y-axis in all graphs is the number of rounds of observation the attacker needs to expose a subset of Alice's recipients.

4.5.1 Degree of Disclosure

It is easier for an attacker to obtain a subset of Alice's recipients than to find all of Alice's recipients. We ran simulations to evaluate how different cover traffic approaches affect the attackers ability to expose a number of Alice's recipients. The graph in Figure 4.1 shows that as the attacker tries to expose more number of recipients, the amount of observation rounds significantly increases. In comparison, Figure 4.2 shows that with more active background senders, the effectiveness of cover traffic is more pronounced. When RB cover is used, the number of rounds sharply increase when more than 70% of her recipients are exposed. When only Alice sends dummies, the rise in number of rounds is more modest when compared to when RB cover is also used. In our remaining experiments, we fix the number of recipients to be exposed at 30% which we simplify to 10 recipients.

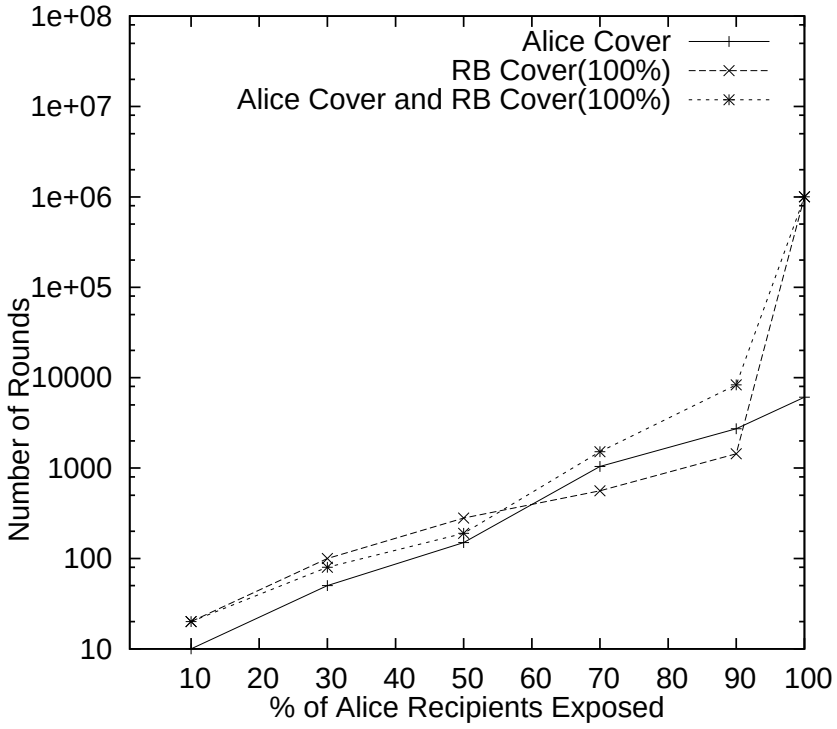

Figure 4.1. Median rounds to identify a subset of Alice's recipients. Background **(BG)** volume = 125 messages/round. Mix delay probability P_{delay}=0.5.

4.5.2 Effect of Background Senders

The graph in Figure 4.3 illustrates the effect of background dummy messages on the number of rounds needed to correctly identify 10 of Alice's recipients. Alice generates dummies according to a geometric distribution. Alice's dummy distribution parameter varies from 0.1 to 0.9 as seen along the x-axis. The effect of background traffic volume (BG) is clearly visible in this graph. When $BG = 125$, the effect of background and Alice dummy messages is very low. In the case when $BG = 1700$, cover traffic has a greater impact. As Alice's dummy volume increases, the number of rounds needed to identify Alice's recipients increases. Further, we see that when the background senders also send cover traffic, it becomes increasingly difficult for the attacker to successfully identify Alice's recipients. When the background senders

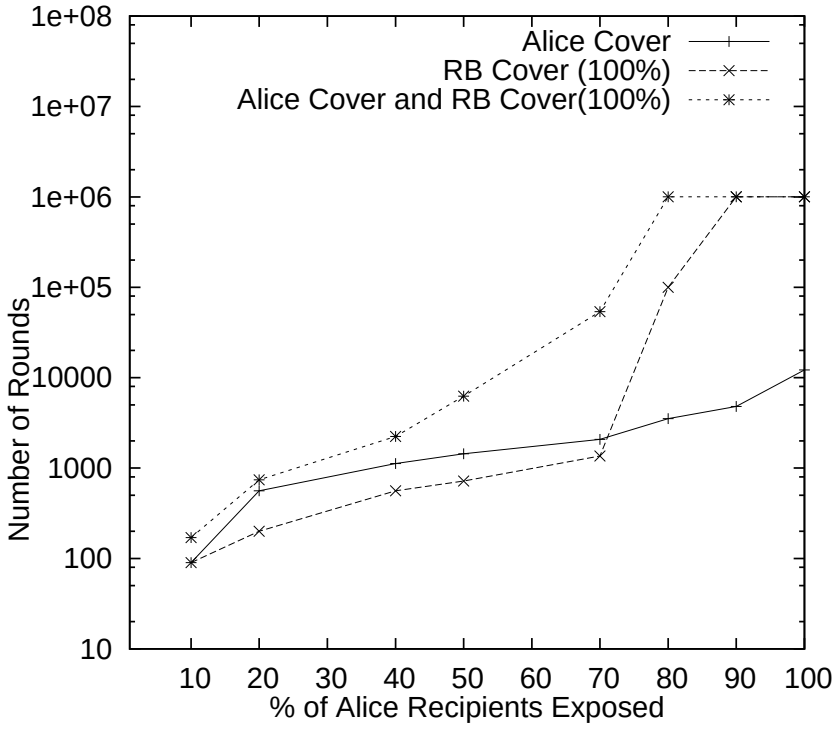

Figure 4.2. Median rounds to identify a subset of Alice's recipients. Background **(BG)** volume = 1700 messages/round. Mix delay probability P_{delay}=0.5.

generate cover traffic at 10% of real traffic and Alice increases her dummy distribution parameter to 0.9, it takes more than one million rounds to correctly identify ten of Alice's recipients.

4.5.2.1 Attacker Adjustment

The attacker can counter the effect of background cover by estimating the number of dummies that the background sends per round. The attacker can observe the number of senders sending per round and has knowledge of their dummy policy. Once the estimate is obtained, the attacker simply has to subtract the number of estimated dummies from the number of observed background messages and continue as if there

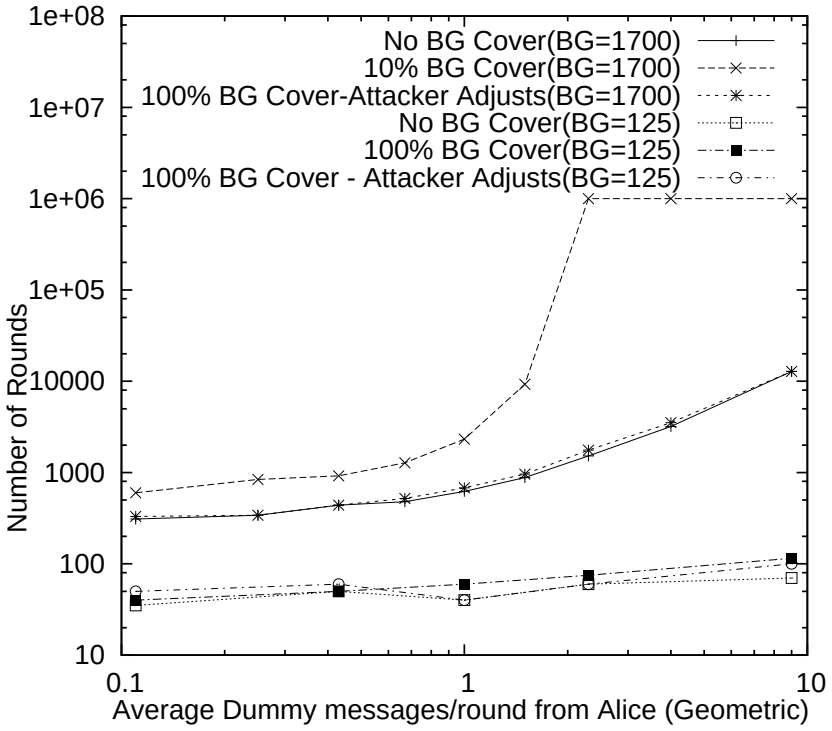

Figure 4.3. Effect of Background Cover and Attacker Adjustment. Median rounds to guess 10 recipients.

were no dummies. Figure 4.3 shows how attacker adjustment can completely negate the effect of background cover, even if background senders use 50% or 100% dummies.

The estimation of total background dummies per round is simple if all senders use the same dummy volume parameter. If senders use arbitrary dummy volume parameters, selected independently or even randomly varied over time, it becomes more difficult for the attacker to estimate the background dummy volume. The attacker could attempt to subtract the average system output from the average system input, as this provides an average of the sum of the background dummies plus Alice's dummies. This suggests another benefit of RB cover traffic, as the attacker would have greater difficulty in measuring the background cover traffic if the number of real messages is hidden in the system output as well. To gain this benefit, a dynamic

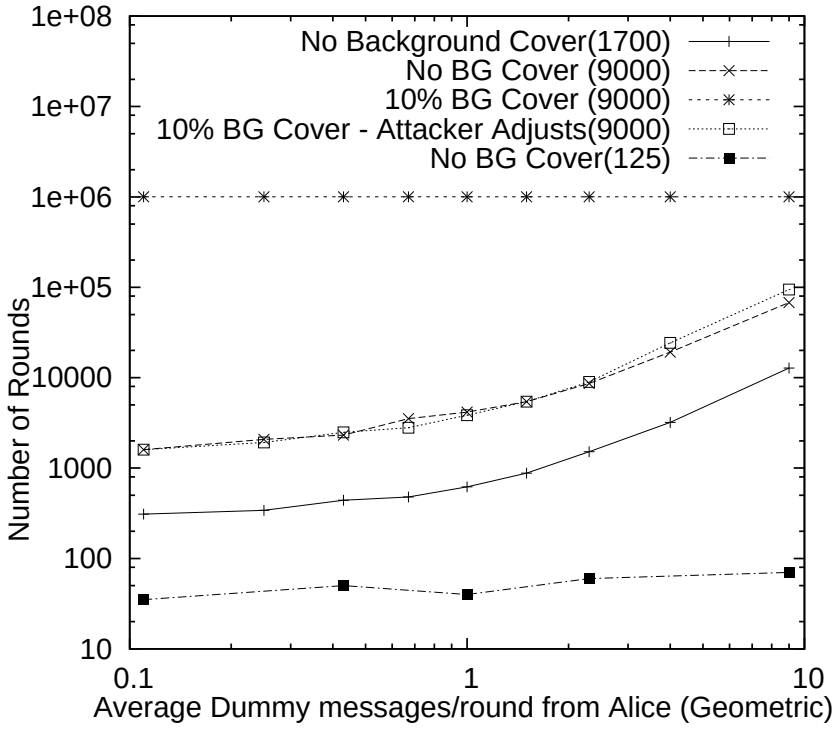

Figure 4.4. Effect of increase in Background Volume. Median rounds to guess 10 recipients.

amount of background cover traffic is required, rather than the fixed percentage of real traffic that we have studied in this paper.

4.5.2.2 Larger Number of Participants

Figure 4.4 shows that as the number of participants in the mix increases, the anonymity of individual participants correspondingly increases. In this simulation we increased the volume of background traffic from a normal distribution with mean 1700 to a normal distribution with mean 9000 messages per round. As observed in the graph, the time for the attacker to expose the same number of recipients more than doubles when participants send messages more frequently.

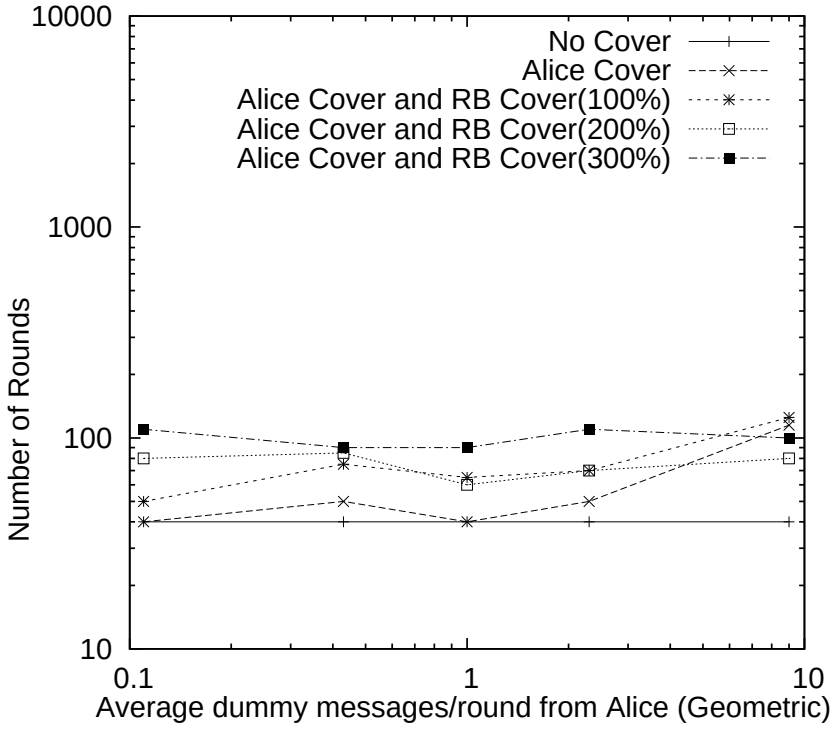

Figure 4.5. Effect of RB cover traffic. Median rounds to guess 10 recipients. Background **(BG)** volume = 125 messages/round.

4.5.3 Effect of Receiver-bound Cover

Figures 4.5 and 4.6 show the effect of RB cover traffic. The mix generates RB dummies equal to the number of real messages per round. We also studied whether the presence or absence of cover traffic from Alice would affect the number of rounds needed to identify Alice's recipients. As Figure 4.6 shows, cover traffic from Alice alone does not have a significant impact on number of rounds. When Alice sends dummies in the presence of RB cover the effects are more pronounced. Compared with Figure 4.5, we see the extent to which increasing the number of background messages helps improve the effectiveness of RB cover. When $BG = 125$, RB cover up to 300% does not significantly degrade the attack.

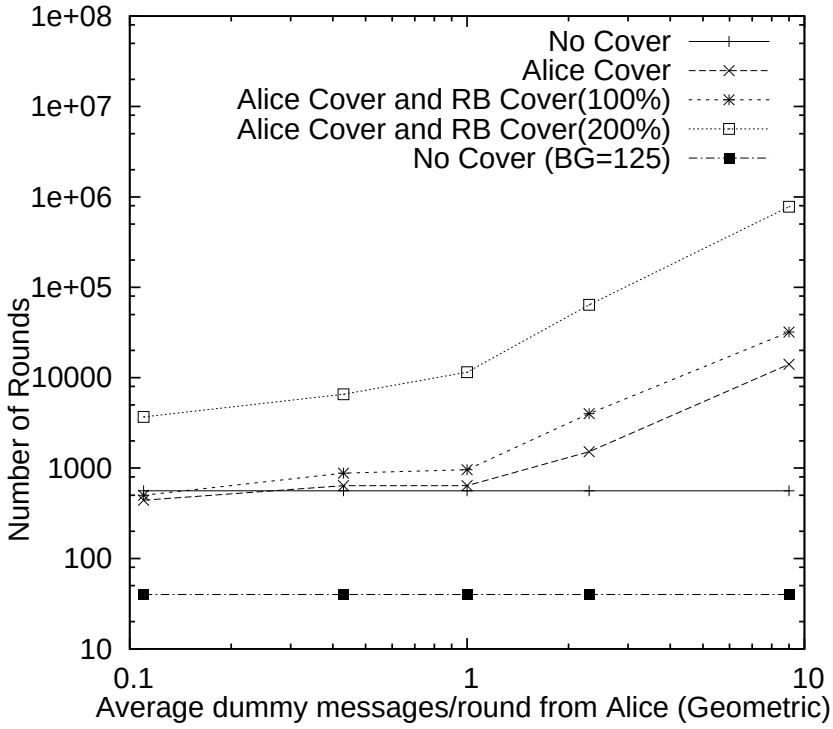

Figure 4.6. Effect of RB cover traffic. Median rounds to guess 10 recipients. Background **(BG)** volume = 1700 messages/round.

Figures 4.7 and 4.8 shows how the increase in delay distribution at the mix makes the attack harder. As before, there is greater benefit in increasing P_{delay} is when the background senders are more active. When the mix exhibits a delay probability higher than 0.5, the number of rounds increases more rapidly. When RB cover is increased to 200% and P_{delay} is more than 0.3, the attack takes more than one million rounds.

4.5.4 Partial Observation

The attack becomes slower when the adversary is a partial observer. Partial observation is a more real situation than a full observer for reasons discussed earlier.

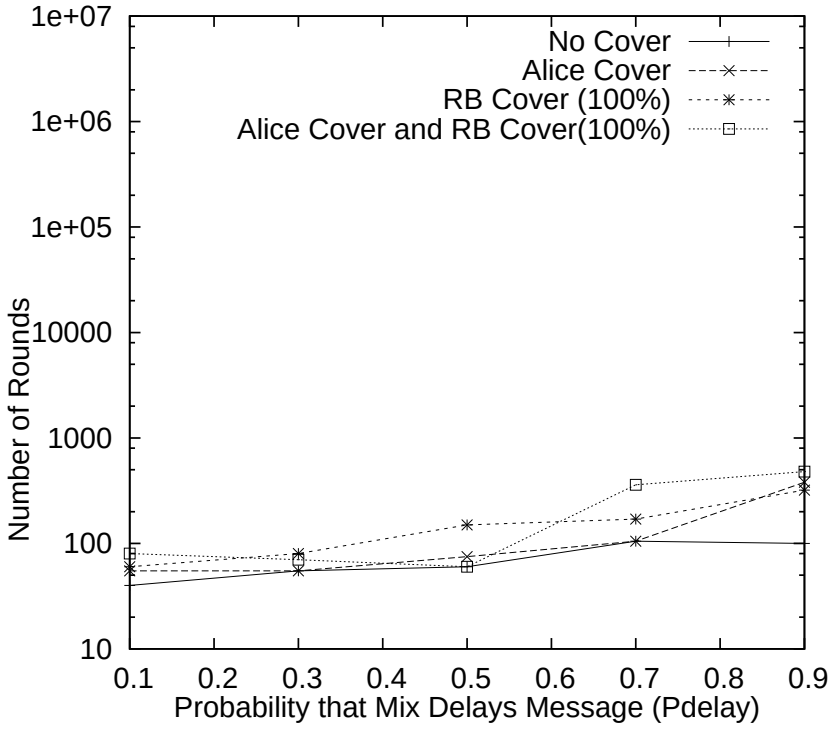

Figure 4.7. Effect of increased delay distribution in the mix. Median rounds to guess 10 recipients. Background **(BG)** volume = 125 messages/round.

Figures 4.9 and 4.10 show the results of a partial adversary who can observer 50% of all traffic going into and out of the anonymity system.

4.6 Discussion

In Section 5.5, we show how RB cover traffic can be used to successfully delay statistical analysis. We now touch upon the implementation aspects that RB cover should exhibit in real-world networks. There are three main considerations:

- Cover traffic must resemble real traffic in order for it to effectively anonymize user traffic.
- Receivers must tolerate the presense of dummy messages.
- The costs of the cover traffic should not be too high for the mixes or the receivers.

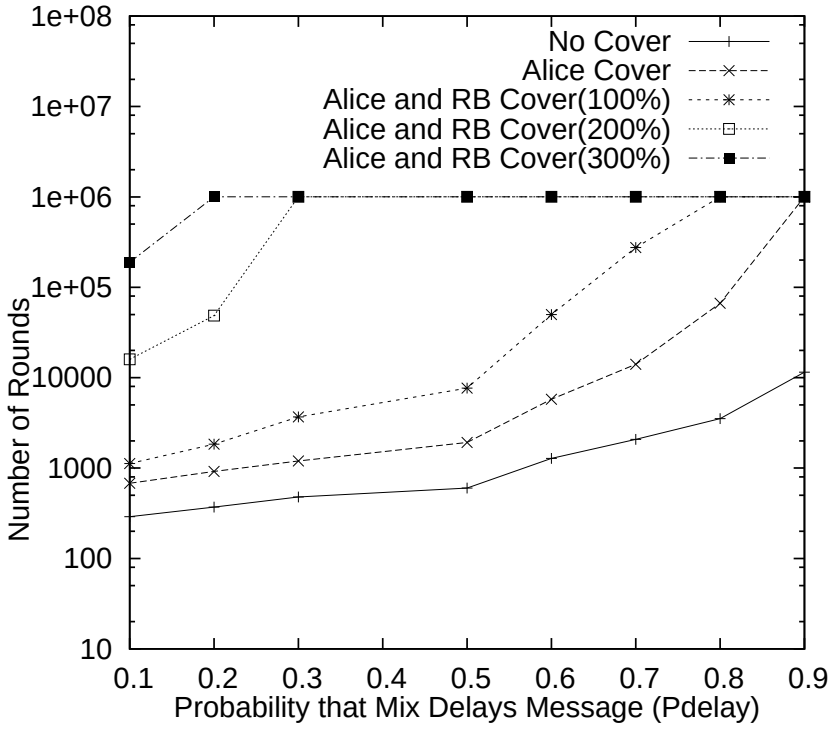

Figure 4.8. Effect of increased delay distribution in the mix. Median rounds to guess 10 recipients. Background **(BG)** volume = 1700 messages/round.

We study these both in the context of high-latency and low-latency mixes, as intersection attacks apply to both types of system. The two forms of cover traffic that we can use are encrypted and unencrypted, each with different advantages and applications.

4.6.1 Encrypted Dummies

Making cover traffic that looks like real traffic is challenging. Content, timing, and receiver selection must all appear to be the same as users' messages. Realistic content is relatively easy to generate if it is encrypted. For high-latency message delivery, such as anonymous email, we can craft packets that appear to be encrypted using PGP [19] or S-Mime [20] but with random payload bytes (in Radix-64). The receiver could attempt to decrypt the random payload and discard the email when

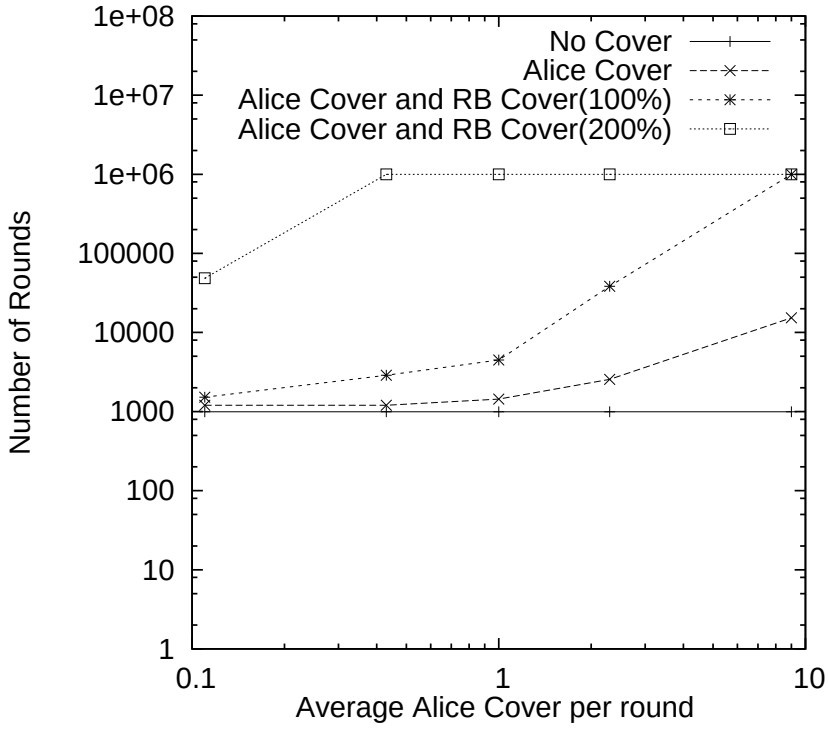

Figure 4.9. **Partial Observer:** Median rounds to guess 10 contacts with increasing Alice cover traffic. Background traffic volume = 1700 messages/round.

it doesn't decrypt properly. There is some cost to the receiver in this case, although email clients could automate this process and remove most of the cost that the receiver actually notices.

One problem with only sending dummies designed to appear encrypted is that, if some of the real messages are not encrypted, the attacker can discount the presence of those encrypted messages. The attacker takes an estimate d' of the number of RB dummies (say, d), based on knowledge of the mixes' distribution of sending those dummies. If the total number of messages is n, and the number of unencrypted real messages is u, which are both measurable, then the chance that any packet with a random payload is a real message is estimated as $p'_{real} = (n - u - d')/(n - u)$. p'_{real} becomes a discounting factor on the additions to vector $\overrightarrow{o}$ in each round. The impact

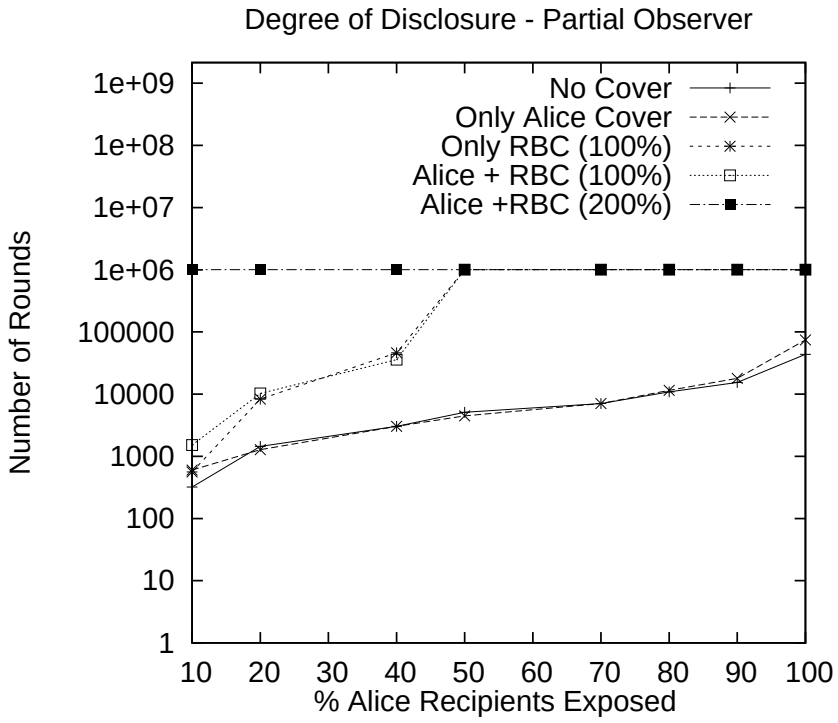

Figure 4.10. **Partial Observer:** Median rounds to guess increasing number of Alice contacts. Background traffic volume = 1700 messages/round.

of this depends on the ratio of encrypted real messages to total real messages. If the ratio is high, we may be able to increase the number of dummies to compensate. If the ratio is low, i.e. there are few real encrypted messages, the attacker can discount much of the cover traffic.

4.6.2 Unencrypted Dummies

As real traffic may also be unencrypted, we propose the use of unencrypted dummies for some applications. There are many applications where users often do not use encryption, including email. In such a case, the mix has to generate cover traffic that carefully replicates real traffic. Messages with randomly-generated payloads would be useless since they can be easily differentiated from real traffic.

For email, messages must be constructed that look like real messages. Messages could be replayed, but the attacker could detect this. The techniques of email spammers could be employed fruitfully here, as copying real text passages, randomization, and receiver customization could all be used to avoid detection by automated systems. Further, the word choice can be designed to match non-spam emails perfectly, as the emails do not need to sell anything. This negates many of the standard Bayesian filtering methods for detecting spam [21, 22, 23]. While attackers could use humans to determine which messages are real, and which are dummies, this would be expensive and might require knowledge about the receiver.

A useful tool to help generate realistic dummies is the behavior of real users. In email, this could mean keeping a record of messages sent to each receiver, and then using this record to help generate new messages with appropriate key words.

4.6.3 Making Receiver-bound Dummies Acceptable

Another critical issue in the use of RB cover traffic is their acceptance by the set of receivers. We have implicitly added some costs to receivers for the privacy of the senders, which may be classified as spam and cause the system to get unwanted negative attention. There are a number of issues and possible solutions which we touch on briefly here.

One way to cast to the problem is to note that RB cover traffic increases the anonymity of the senders connecting to the receivers. It is in the interest of anonymity for these users, so a receiver should allow anonymity networks to send cover traffic to it. Receivers who don't wish to help provide anonymous communications can block messages from the system. Some recipients block connections coming from anonymity systems like Tor [24] exit nodes. We could publish a 'White List' of servers that allow

connections from the anonymity systems, so users can connect to those services via systems like Mixminion [25].

Another way to see the issue is in the light of spam. Today we see that a large percentage of network traffic consists of spam messages [26]. Receivers have developed a number of effective ways to drop or ignore spam messages. RB cover traffic would be a tiny addition to the millions of unwanted messages that flood the network. Further, these unwanted messages help enhance sender and receiver anonymity. Reciever-bound cover would be a small price to pay for the greater benefit of anonymity that it provides to network users. In some cases, especially in Web-browsing, the extra traffic could generally go unnoticed.

Anonymity systems have become popular over the past few years and the number of users participating these systems is continuing to grow. Currently, however, these users remain a small part of the global Internet community. The volume of traffic exiting anonymity systems is low as compared to non-anonymous traffic in the network. RB cover traffic generated to anonymize this fraction of Internet traffic would hardly burden the massive network resources that are in place.

CHAPTER 5

THE REVERSE STATISTICAL DISCLOSURE ATTACK

5.1 Introduction

Statistical disclosure is a well-studied technique that an attacker can use to uncover relations between users in mix-based anonymity systems. Prior work has focused on finding the receivers to whom a given targeted user sends. In this part of the thesis, we investigate the effectiveness of statistical disclosure in finding all of a users' contacts, including those from whom she receives messages. To this end, we propose a new attack called the Reverse Statistical Disclosure Attack (RSDA). RSDA uses observations of all users sending patterns to estimate both the targeted user's sending pattern and her receiving pattern. The estimated patterns are combined to find a set of the targeted user's most likely contacts. We study the performance of RSDA in simulation using different mix network configurations and also study the effectiveness of cover traffic as a countermeasure. Our results show that that RSDA outperforms the traditional SDA in finding the user's contacts, particularly as the amounts of user traffic and cover traffic rise.

Mix-based anonymity systems [2] provide privacy by keeping eavesdroppers from linking communicating parties. Long term intersection attacks are particularly effective in reducing user anonymity in such systems. The most well known practical *traffic-confirmation* attack on mix systems is the Statistical Dislosure Attack [3] in which the attacker targets a single user with the aim of exposing the user's communication partners.

In the traditional form of this attack, the attacker eavesdrops on messages from senders to the mix and messages from the mix to receivers. The attacker uses the frequency of communication between parties to expose links between participating users. The aim of the attacker is to expose the contacts of a target user. Replies and other traffic sent to the targeted user is not considered. In reality, much of the communication in the Internet is two-way. The attacker, often assumed to be a *global eavesdropper* that can see all messages, would likely attempt to extract information from the patterns of traffic sent to the targeted user to learn more about her behavior.

5.1.1 Contributions

We explore how the attacker could extract information from other users' sending patterns to learn more about the target user and her contacts. In particular, we introduce a new attack called the Reverse Statistical Disclosure Attack (RSDA) (§5.3). In the RSDA, the attacker simply applies the SDA to each user who sends messages. Some of the contacts of the targeted user — henceforth, we will refer to her as Alice — can be guessed based on the SDA applied to Alice. Additionally contacts of Alice can be guessed by examining the SDA results of other users. Let us consider Alice's friend Bob, who have replies regularly to Alice's messages or may simply send new messages to Alice. The attacker applies the SDA to Bob, and may be able to guess that Alice is one of his receivers. The RSDA leverages this information to note that Bob is a likely contact of Alice, even if the SDA did not allow the attacker to identify Bob as a receiver of Alice.

Note that the RSDA has a different model of what the attacker is interested in (§5.2), compared with the SDA. In the SDA, the attacker is only interested in the receivers to whom Alice sends. In the RSDA, the attacker wants to know all of the contacts with whom Alice communicates, whether sending or receiving. We

believe that this is more realistic; traffic analysis is not generally confined to finding relationships in one direction.

The way RSDA uses information gained about other senders to learn about Alice is unique. In particular, we know of two other approaches that use similar information: the Two-Sided SDA (TS-SDA) [6] and the Perfect Matching Disclosure Attack (PMDA) [5]. We discuss these in more detail in §**??**, but briefly point out the key differences here. The TS-SDA assumes that the attacker is only interested in receivers to whom Alice *initiates* a message and attempts to filter out the statistical influence of Alice's replies on her SDA values. This is the opposite assumption from the RSDA model, in which the attacker is interested in any contacts of Alice, whether Alice initiates messages to them or not. The PMDA compares Alice's sending behavior to other senders' behavior with the intention of matching the senders to their most likely receivers in each *batch* of messages. PMDA is not looking for senders to Alice; RSDA is.

We use detailed simulation (§5.4) to study RSDA using different mix network configurations. Cover traffic has been recognized as an effective way to counter Statistical Disclosure Attacks [14, 27]. Hence, we also study the effectiveness of cover traffic, including *background cover* and *receiver-bound cover*, as a countermeasure. Our results (§5.5) show that RSDA outperforms SDA particularly as the amounts of user traffic and cover traffic increase. Cover traffic from Alice affects SDA adversely and increases the time to 900 rounds; an increase of over three times compared with no cover. RSDA is extremely resilient to user cover and succeeds in only 250 rounds with cover and 100 rounds when no cover is present. We also found that as the total number of messages mixed in each round increases, both SDA and RSDA need more time to succeed. However, RSDA takes close to half the number of rounds compared to SDA as the mix batch size increases from 100 to 500 messages. When a binomial

mix, having a more complex mixing strategy than the threshold mix is used, RSDA still proves to be a much faster attack compared to SDA. Furthermore, in the presence of increasing Alice cover, RSDA increases from 1000 rounds to only 1800 rounds while the increase in time for SDA is almost four times more going from 3000 to 6000 rounds with increasing Alice cover. RSDA is also affected very little in the presence of receiver-bound cover traffic. We conclude that RSDA is a much speedier attack than the traditional SDA. It shows a sizeable improvement over SDA and achieves high performance even in the presence of counter-measures like user and receiver-bound cover traffic.

5.2 Model

We now describe a model for our study of RSDA. We start by describing how we model mixes and users' communication patterns, and then we discuss our attacker model.

5.2.1 Mixes

We investigate statistical disclosure attacks against a simplified model of mixes. We use the term *mix* to refer to the entire mix network or mix cascade and abstract away details such as the number of mixes and their configuration. All users send their messages and cover traffic to the mix, and the mix sends messages on to all the receiving users.

We investigate RSDA's effectiveness against two types of mixes:

Threshold Mix. The threshold mix [2] collects a fixed number B (the batch size) of input messages before relaying the messages in a random order en route to their destinations. Each cycle of input and output together is called a *round*. **Binomial Mix.** In a binomial mix [28], each incoming message is subject to a biased coin toss

to decide whether the message leaves the mix in the current round or is delayed until a later round. The mix uses P_{delay} as the delay probability to bias the decision.

5.2.2 Communication Patterns

As we study the effectiveness of statistical attacks based on profiling users, the communication patterns of the users are critical to our evaluation. The three main features of the model are contacts (who sends to whom), sending behavior (how often does each user send to each of her contacts), and cover traffic.

We assume that there are N users, and we use a uniform model for establishing contacts between them. Specifically, each user, including Alice, has a fixed number of receivers m. The receivers are chosen uniformly at random from the set of other users. Unlike prior work in statistical disclosure attacks [14, 6, 5], we do not have separate sets of senders and receivers. Rather, each user will be a receiver for some of the other users. All of the users that communicate with a given user are included in that user's *contacts*. The total number of contacts per node will vary, but will be $2m$ on average.

Since the attacker focuses on a targeted user, Alice, we distinguish between Alice's behavior and other users' behavior. Alice sends n_A messages in a given round. n_A is a random variable selected from a Poission distribution with average rate λ_A. Alice chooses the recipients of her messages uniformly from her set of contacts. Users other than Alice are called *background senders*. When the mix uses a fixed batch size as in the case of a threshold mix, background senders together send $n_B = B - n_A$ messages. If the batch size is variable, as in the case of a binomial mix, background senders together send n_B messages, where n_B is chosen from a normal distribution with mean μ.

Cover traffic consists of fake messages called *dummy messages* that are inserted into the network along with real messages. Dummy messages are meant to look like real messages and cannot easily be distinguished from real messages. Usually, this means that the content of real messages that would be encrypted is replaced with random bits. The receiver of the dummy messages can recognize that they are fake, as they do not decrypt properly, and drops such messages on arrival. In our model, we use two types of cover traffic for the simulations. *Alice cover* consists of dummy messages that Alice sends to the mix. These messages are dropped at the mix. In each round in which Alice participates, she inserts zero or more dummy messages along with real messages. Alice may send dummy messages with no real messages in some rounds. *Receiver-bound cover* (RBC) consists of dummy messages from the mix to receivers. See [27] for details on how RBC is used to counter SDA.

5.2.3 Attacker Model

We model the attacker as a global eavesdropper who can observe all links from senders to the mix and all links from the mix to recipients. The target of the adversary is Alice and the adversary's aim is to determine with whom Alice communicates, i.e. to identify her contacts. The attacker observes all communications into and out of the mix during a number of rounds, including rounds with and without Alice's participation . The attacker observes only the incoming and outgoing links from the mix and does not observe activity inside the mix. This assumption is for the simplicity of the model, as there are many configurations for a mix network, but also because SDA and RSDA are effective without observations of activity inside the mix network.

Mathewson and Dingledine developed a simulation, including the use of a binomial mix (called a pool mix in their paper), to investigate the effect of a number of parameters on the performance of SDA [14]. They found that as the number of

Alice's contacts grew, the rounds of observation to expose her full contact list correspondingly increased. They also found that cover traffic from Alice was effective in slowing, but not preventing, SDA. Cover traffic from Alice was found to be more effective when the delay probability of the binomial mix was increased. Increasing the mix delay spreads out the incoming traffic over a number of outgoing rounds, making it more difficult for the attacker to estimate which set of receivers might have gotten the messages from Alice.

5.3 Reverse Statistical Disclosure Attack

In the Reverse Statistical Disclosure Attack (RSDA), the attacker first applies the SDA to all N users. The attacker learns two pieces of information from this step. First, the attacker applies the SDA to Alice to learn about to whom Alice sends messages. Second, by applying the SDA to other users, he can determine which of them send to Alice. The attacker then combines this information to find the most likely contacts of Alice.

We break up the attack into three parts: (1) *forward observation*, or observations of Alice's sending behavior; (2) *reverse observation*, observations of other users' sending behavior; and (3), combining forward and reverse observations.

5.3.1 Forward Observation

In each round of observation the attacker records information in the forward direction as described in Section 2.3. This allows the attacker to calculate D_A, a set of scores representing Alice's estimated sending behavior.

5.3.2 Reverse Observation

In each round of observation the attacker also records information in the reverse direction. For a user X, the attacker records n_X, the number of messages sent by X, n_B, the number of messages sent by other users, and $\overrightarrow{o}$, the distribution of messages received by users in rounds that X sends. The eavesdropper also records D_N^X, the distribution of of messages received by users in rounds that X does not send. Using these observations, the eavesdropper does the SDA on X by using the following equation:

$$\overline{O} = \overline{n_X}.D_X + \overline{n_B}.D_N^X \tag{5.1}$$

With these observations, the attacker can apply Eqn. 5.1 to estimate D_X, the scores representing X's sending behavior.

Now let $D_X[A]$ represent the attacker's estimate of user X's sending behavior to Alice. We create a new vector D_R, such that $D_R[X] = D_X[A]$. In other words, D_R represents the estimated sending behavior of all other users with respect to Alice.

5.3.3 Combining Observations

The RSDA estimate of Alice's most likely contacts, $\widehat{D}_A$, can be determined by combining D_A and D_R calculated from the forward and reverse observations, respectively. D_A and D_R are combined by first normalizing and then obtaining a weighted mean of the two distributions. If v_f is the volume of traffic observed in the forward direction and v_r is the volume of traffic in the reverse direction, then we obtain:

$$\widehat{D}_A = \frac{v_f.D_A + v_r.D_R}{v_f + v_r} \tag{5.2}$$

Note that we could keep the information separate and simply determine Alice's receivers and those who send to Alice in isolation of each other. However, Alice's

receivers will reply to her and vice versa. Since we assume that the attacker is interested in all of Alice's contacts, combining the information helps him learn more.

To see this, let us consider two users, Bob and Carol. Bob is a contact of Alice who occasionally sends to Alice and receives replies, while Carol is not Alice's contact. Over a very large number of rounds, the SDA alone will distinguish between Bob and Carol with respect their contact with Alice. In fewer rounds, however, Bob and Carol may have very similar statistical links to Alice. Since Alice replies to Bob, combining their SDA observations should provide better evidence that they are contacts. On the other hand, since Alice never sends to Carol, combining their SDA observations will likely weaken the evidence for them being contacts. Thus, combining scores should improve the relative evidence for real contacts.

5.4 Simulation Setup

We simulated the process of sending and receiving messages via a mix network according to the model described in Section 5.2. The parameters used in our simulations are discussed in this section and summarized in Table 5.1. The number of users in the system N is set to 100. The number of contacts for Alice is $m = 20$. The simulations were carried out for the two attacks that we are comparing: SDA and RSDA.

5.4.1 Mix Behavior

- Threshold Mix: For the threshold mix we set the batch size $B = 200$ messages a round.
- Binomial Mix: For the binomial mix, the probability that an incoming message is delayed is set to $P_{delay} = 0.2$.

Table 5.1. Simulation parameter values

Parameter	Value	Description
N	100	Number of users in the system
m	10	Number of Alice's contacts
B	100 to 500	Batchsize of threshold mix
P_{delay}	0.1 to 0.9	Probability of delay of binomial mix
P_{reply}	0.5	User's reply probability
λ_A	5.0	Alice message initiation rate i.e. messages/round
λ_U	1.0 to 10.0	User message initiation rate i.e. messages/round per user
λ_{A_d}	1.0 to 10.0	Alice dummy initiation rate per round
$RBCVOL$	$10\% to 100\%$	RBC volume as per cent of real messages/round
$CUTOFF$	10^5 10^6	Simulation cutoff, Threshold Mix Simulation cutoff, Binomial Mix

Figure 5.1. Median rounds to identify a 50% of Alice's recipients. Threshold mix with no cover traffic.

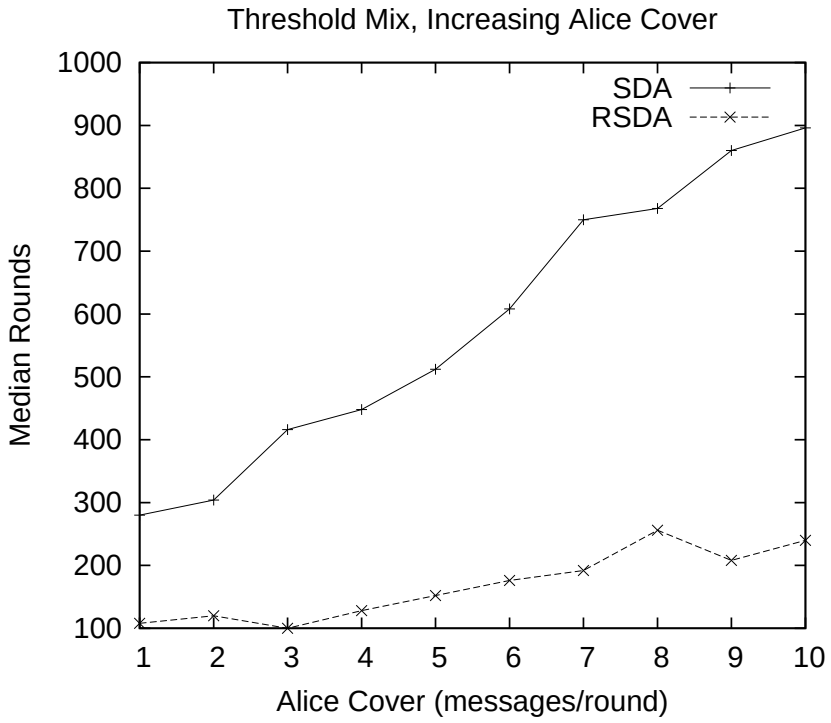

Figure 5.2. Median rounds to identify a 50% of Alice's recipients with Alice Cover. Threshold mix with $B = 200$.

5.4.2 Message Generation

- Alice Initiation: The number of messages Alice initiates is based on a poisson distribution with an average rate of $\lambda_A = 5.0$ messages per round.
- Other Users Initiation: The number of messages sent by users apart from Alice is based on a poisson distribution with an average rate of $\lambda_U = 5.0$ messages per round.
- User Reply Behavior: Users, including Alice, reply to messages they receive from other users with a probability of $P_{reply} = 0.5$. If users decide to reply, they do so in the very next round.

5.4.3 Cover Traffic

- Alice cover: The number of dummy messages per round is determined using a Poisson distribution with rate λ_{A_d}, which is varied from 1.0 to 10.0 messages per round for our simulations.
- Receiver-bound Cover: For the threshold mix simulations, the volume of receiver-bound cover is set to $RBCVOL = 100\%$. This means the number of dummy messages sent from the mix to users per round is 100% of the number of real outgoing messages from the mix to users in that round. For the binomial mix simulations, the volume of receiver-bound cover is set varied from $RBCVOL = 10\%$ to $RBCVOL = 90\%$.

5.4.4 Measuring Attacker Success

The attacker eavesdrops on communications between users over a period of time that is divided into rounds. We use the median number of rounds for the attacker to find 50% of Alice's recipients as a measure of the attacker's success. In [27] we discuss why exposing a fraction and not all of Alice's contacts sufficiently degrades her anonymity. The number of rounds of attacker observation is bounded by a *CUTOFF* value, so that the simulation can end in finite time when the attack does not converge. The observation CUTOFF is set to 10^5 when the median rounds to identify Alice's contacts is lower than 50000 rounds. The CUTOFF is set to 10^6 rounds when the median rounds is higher. Generally, we observed lower median rounds for the threshold mix and higher median rounds for the binomial mix.

5.5 Results

In this section we discuss the results of our simulations. Please note the use of a logarithmic y-axis in some graphs.

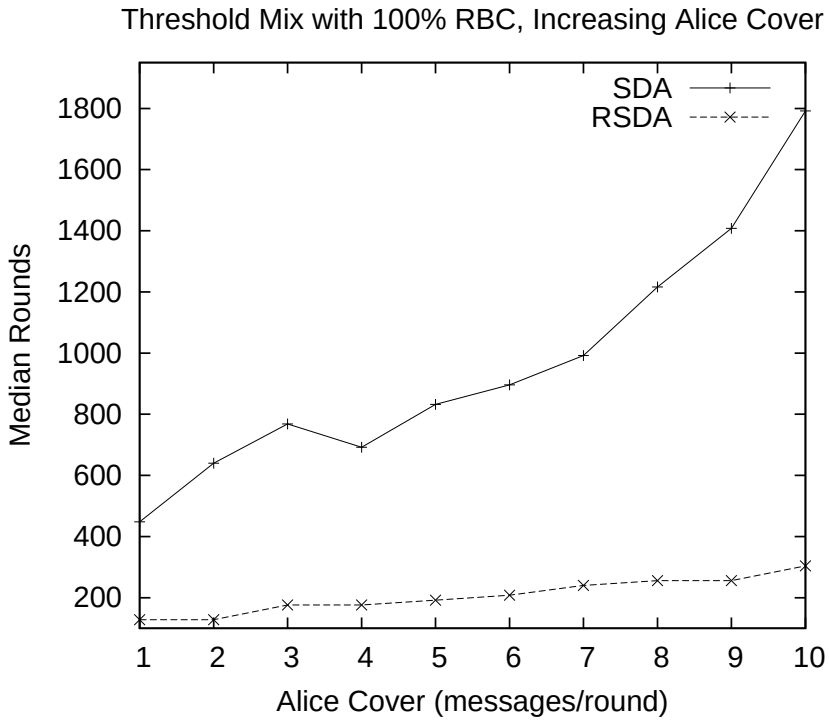

Figure 5.3. Median rounds to identify 50% Alice's recipients. Threshold mix with $B = 200$, $RBCVOL = 100\%$.

5.5.1 Simple Threshold Mix

5.5.1.1 No Cover Traffic

We ran multiple simulations to compare the performances of SDA and RSDA. We studied the effectiveness of the attacks for different batch sizes ranging from $= 100$ to 500. Alice sends at a rate of $\lambda_A = 5.0$ messages per round. The results are shown in Figure 5.1. We see that when a threshold mix is used, RSDA outperforms SDA especially for higher batch sizes.

5.5.1.2 Alice Cover

For the next simulation we fixed Alice's message initiation rate and other user's message intiation rate at, $\lambda_A = \lambda_U = 5.0$ messages/round. Alice sends cover traffic

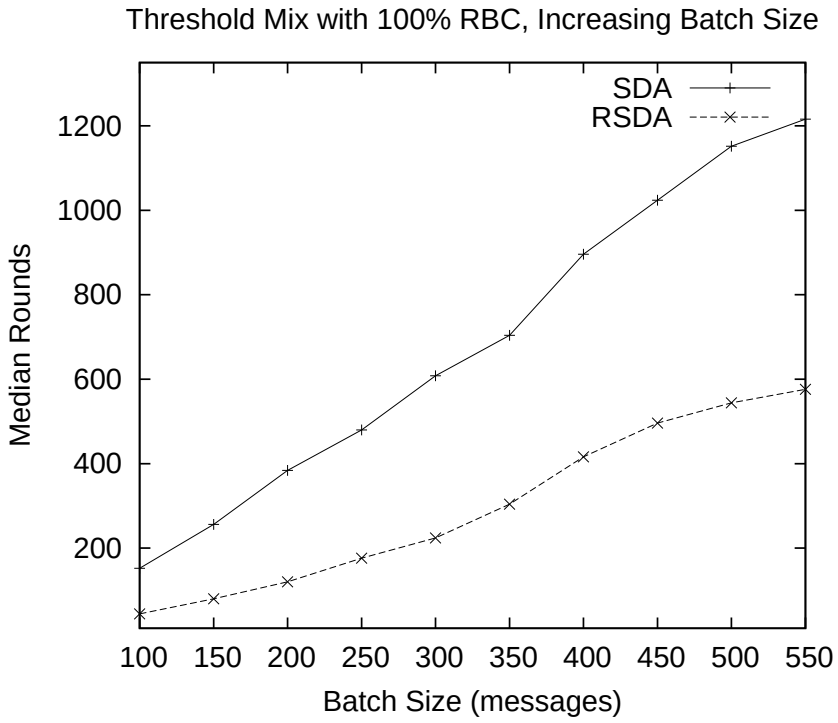

Figure 5.4. Median rounds to identify 50% Alice's recipients. Threshold mix, $RBCVOL = 100\%$.

increasing from 1.0 to 10.0 messages/round. Figure 5.2 compares the performance of SDA and RSDA in the presence of increasing cover traffic from Alice. For 10.0 messages/round of Alice cover, the number of rounds for SDA increases by a factor of three to 900 rounds. RSDA on the other hand is able to perform well even when Alice cover twice her real message rate and remains below well 250 rounds.

5.5.1.3 Receiver-bound Cover

In addition to Alice cover we added receiver-bound cover with $RBCVOL = 100\%$ and compared the performance of SDA and RSDA. The results are shown in Figure 5.3. The median rounds for attacker success with SDA goes to about 1800 rounds when Alice cover is $\lambda_{A_d} = 10.0$ messages/round. In the presence of RBC and

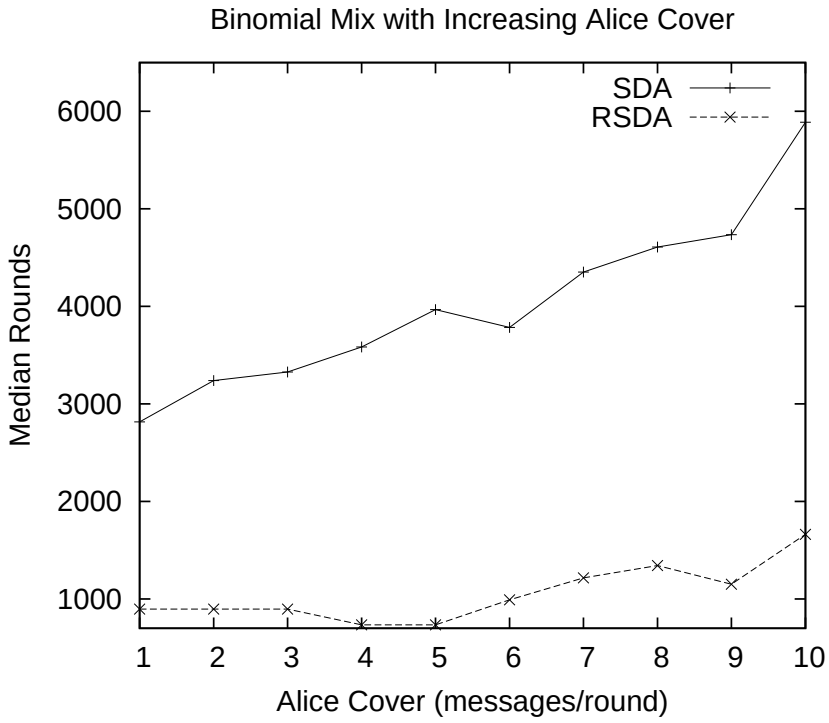

Figure 5.5. Median rounds to identify 50% Alice's recipients. Binomial mix with increasing Alice cover.

high volume of Alice cover, RSDA is still able to expose 50% of Alice's contacts within about 300 rounds which is 6 times lesser than SDA.

We also studied the performance of RSDA when Alice cover is not used and only the mix sends RBC to users. The results of this scenario is shown in Figure 5.4. SDA and RSDA are compared for increasing values of mix batch size.

5.5.2 Binomial Mix

5.5.2.1 Only Alice Cover Traffic

In our next simulation we compared the performance of SDA and RSDA using a binomial mix. The results are shown in Figure 5.5. We see that, like in the threshold case, RSDA outperforms SDA. When Alice sends one dummy message/round, RSDA

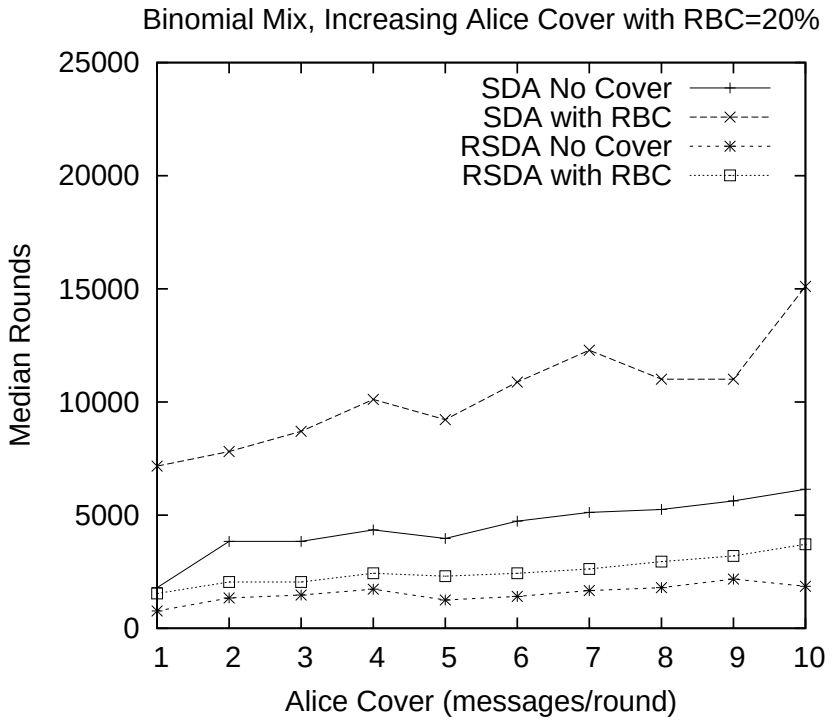

Figure 5.6. Median rounds to identify 50% Alice's recipients. Binomial mix with RBC=20%.

succeeds in a third of the time taken for SDA to succeed. At higher volumes of Alice cover, RSDA continues to succeed faster than SDA.

5.5.2.2 Alice and Receiver-bound Cover

In this simulation we show the impact of introducing RBCVOL=20% along with increasing Alice cover. We compare the performance of both the attacks when the mix does and does not send receiver-bound cover. The results are shown in Figure 5.6. We see that the time needed for SDA more than doubles in the presence of 20% RBC. RSDA on the other hand is not affected by RBC to the same degree as SDA.

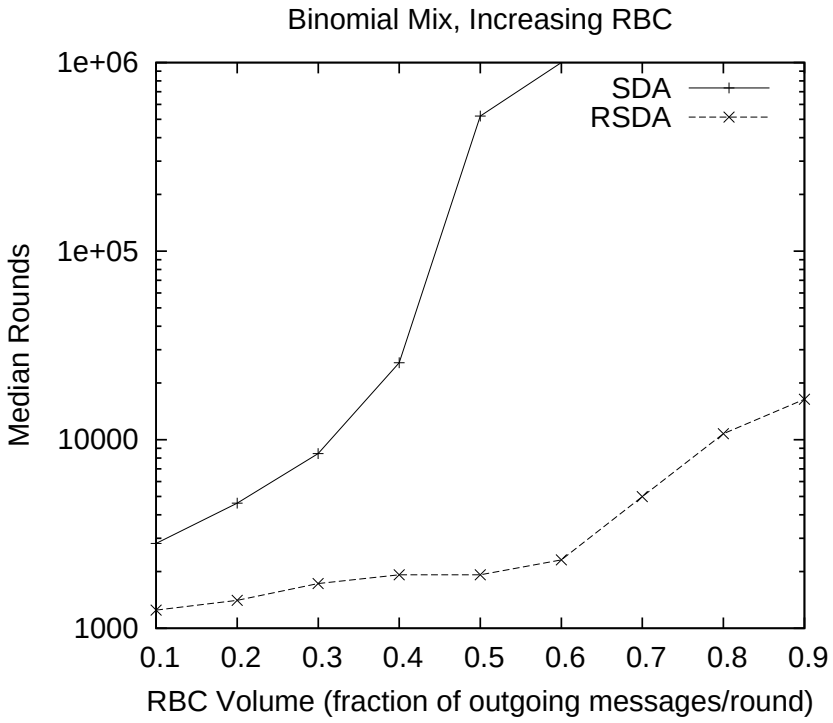

Figure 5.7. Median rounds to identify 50% Alice's recipients. Binomial mix with increasing RBC.

5.5.2.3 Only Receiver-bound Cover

We study whether higher amounts of RBC affect the performance of RSDA and compare the results with the performance of SDA. In order to understand how RBC affects RSDA, we set Alice cover to zero and increased the volume of RBC generated by the mix from $RBCVOL = 10\%$ to $RBCVOL = 90\%$. The results are shown in Figure 5.7. We see that as the amount of RBC increases the taken for SDA dramatically increases from 2816 rounds to over a million rounds. RSDA shows a ten-fold increase from about 1000 rounds to 10000 rounds of observation when RBC is increased from 10% to 90%. However, compared to SDA, RSDA is significantly more tolerant to receiver-bound cover traffic.

CHAPTER 6

SHAPING NETWORK TOPOLOGY FOR PRIVACY AND PERFORMANCE

Mix-based anonymous systems consist of a network of mixes that are used to relay user traffic. If a user Alice, wants to communicate with another user Bob via the mix network, she first chooses a sequence of mixes over which to route her messages. Alice is free to pick whichever mixes she wants to include in her path, starting with herself and ending in the final mix, such that for the duration of the connection her messages are routed over this path.

6.1 Restricted- route Mix Networks

6.1.1 Benefits

Current mix networks use a complete graph topology. Users who wish to relay their messages through the mix network can randomly choose any set of nodes to route their messages via the mix network. In [29] the authors show that free-route mix networks are vulnerable to *intersection attacks* especially if many of the mixes in the network are compromised. If all mixes except a single mix in the path of a message are compromised, an attacker can use intersection attacks to reduce the anonymity set Alice. If Alice always uses the same sequence of mixes to send all her messages, the attack becomes easier. One way to protect against such attacks is to not allow users to freely choose each mix of the path their messages will take.

6.1.2 Approaches

6.1.2.1 Mix Cascades

[29] suggests that users pick from a set of predetermined paths called *cascades of mixes*. Each cascade is a sequence of mixes, with equal number of mixes in every cascade. This paper argues that if larger amounts of traffic from different users are mixed at each hop, the attack becomes more difficult. For a given amount of traffic passing through the mix network, concentration of traffic at a mix is possible by restricting the number of possible routes through the network. One way of achieving restricted routes is by using mix cascasdes.The AN.ON project uses this type of architecture.

6.1.2.2 Sparse Network Topology

Danezis [30] proposes that a mid-way between free-route mix network and mix cascades. He suggests that a restricted route mix network can be based on the topology of a sparse graph. The user can freely select a starting node. Subsequent nodes along the path are chosen according to the sparse graph topology of the network. He suggested the expander graph topology be used for the mix network topology. A D-regular random graph is an expander with high probability, where D is the number of neighboring nodes for a given node. A D-regular graph is a graph in which every node has connected to exactly D other nodes. The fewer edges in the expander topology helps to concentrate traffic entering at a given mix by shaping it flow through fewer mixes instead of dispersing traffic through many mixes as in a complete graph topology. Another advantage is that the small and constant degree D, means that each mix connects to only D other mixes as compared to $N-1$ mixes in a fully connected network thus reducing possible routes in the network.

A lower degree per mix reduces the amount of cover traffic that is needed counter traffic analysis attacks and attacks described in [15]. However, mix cascades have some disadvantages. For example, it is easy to perform DoS attacks against cascades because disabling even one node of the cascade makes the cascade unusable. Scalability is also an issue.

6.2 Expander Graphs

6.2.1 Expander Properties

Expander graphs are sparse graphs that show high connectivity. These graphs are common in the natural world and can be used to model social networks, relationships between species, organizations etc. In this part, we investigate the use of expander graph topology to improve the security and peformance of mix based anonymity networks.

A graph $G = (V, E)$ in which every vertex has exactly D neighbors is called D-regular graph. A D-regular graph is an (A, K)-expander if for every subset $S \subseteq V$ of vertices in G, such that $\mid S \mid \leq K$, then $\mid N(S) \mid > A \mid S \mid$. Here $\mid S \mid$ is the number of vertices in S and $\mid N(S) \mid$ is the number of vertices in S' that share an edge with any vetrex in S.

This means that expanders contain a relatively small number of edges per vertex D compared to a complete graph with the same number of vertices. This property can be leveraged for providing better security in mix networks.

6.2.2 Expander Construction

A random expander graph can be generated by using distributed algorithms or using the popular zig-zag method. The edges in a random expander are chosen

based on the expansion properties the edge provides to the resulting graph. The edges themselves carry no weight.

In a random expander topology, the number of hops H for a message to mix completely is $O(logN)$, where N is the number of vertices in the network. We generate a random expander and assign latencies to each link. The latencies are obtained from the KING dataset [31]. We run simulations to generate random paths of length $H = logN$ and measure the mean latency to determine the performance. We propose to simulate intersection attacks on nodes in the network and measure the number of rounds to the attack to be successful.

6.3 Security

We measure the security of a complete topology against intersection attacks in terms of the number of rounds of observation needed for the attack to succeed.

6.3.1 Random Walk on an Expander Graph

Picking a series of nodes that form a path through an anonymous network can be likened to a random walk. A random walk consists of visiting a series of nodes one after another, beginning from a chosen starting point. The next node on the path is chosen randomly from the list of neighbors of the current node. The probability of chosing a particular neighbor hop from the set of neighbors may be the same as choosing any other neighbor, in which case we say its is a uniform distribution. Some neighbors might be more likely to be chosen in which case we call it a biased distribution. Mathematically, a random walk can be modeled using a markov process.

6.3.1.1 Markov Process

A markov process consists of a set of predefined states and a set of transition probabilities between each of these states. For the process of choosing a route through a mix-network, the predefined states are equivalent to the nodes in the network and the transition probabilities are equivalent to the connections between the nodes.

6.3.1.2 Equilibrium Distribution

After sufficient number of hops through the graph, the random walk converges to a stationary distrubution. Once the stationary distribution is reached further hops do not result in a change in the distribution.

6.3.2 Entropy

We use the security metric described in [30] to measure the security that a given network toplogy provides. The metric used in that paper is the number of rounds of observation needed to perform the intersection attack.

While an expander topology provides some defense against intersection attacks, it also requires messages to travel more number of hops inside the mix network in order to mix fully. This means, if a message enters the mix network at a particular entry node it needs a minimum number of hops, H, in order to arrive at a random node that does not depend on the entry node. For a complete graph, $H = 1$ because it is fully connected. For an expander graph with fewer edges, $H > 1$. The degree, D, of the graph is inversely proportional to H; as D decreases H increases. As the value of H increases the latency of user connections increases. Latency is measured as the time taken for a message to travel from the sender, Alice, to the receiver, Bob.

We use latency as a metric to measure the performance tradeoff when an expander topology is used.

6.4 Performance

Reputation systems can be used to choose in favor of nodes that offer more reliability and better performance. In [32] the authors describe a link-based path selection strategy that chooses in favor of higher performing links. [33] suggests a way to pick mix nodes based on node-performance in a way that allows users to tune their node choices to trade-off between higher performance and higher security.

We measure the performance in terms of the round-trip time taken for the messages to travel from the entry node to the exit node. The latency of the connection is directly proportional to the round-trip time of the connection.

6.4.1 Latency-based Expander Topology

We propose a latency-based expander topology to achieve the security benefits of using an expander topology as well to achieve improved performance. If the edges for the expander topology are based on latency information, then the resulting expander will have more higher performing links than lesser performing links. Due to this, a randomly chosen path through a latency-based topology is likely to have a better performance.

The trade-off for having a better performance is that the topology generated by biasing edge selection to better performing links may not mix fully in $O(logN)$ hops. This is because, the topology may not be a true expander. In our proposed work we investigate the impact a latency-based expander topology on the security and performance of the mix network. We study how much tradeoff the tradeoff between

security and performance and compared our results between fully connected, random expander, and latency-based expander topologies.

6.5 Simulation

We tested our hypothesis using simulation of a mix network. We used the distributed method of expander construction described in [34] to generate the mix network topology. This method uses an initial set of three nodes that are connected to each other with $D/2$ hamiltonian cycles where D is the degree of each node in the graph that is generated by this method.

We used the KING dataset [31] for simulating the round trip time $RTTs$ between nodes in the network.

6.6 Results

In our first simulation we measured the minimum degree at which maximum entropy can be reached for an n-hop path. In figure 6.1 we compare the minimum degree to reach maximum entropy for a 3-hop and 4-hop path through the anonymous communication system. We measure the minimum degree for increasing number of total nodes in the network, starting from $N = 100$ to $N = 1700$. We found that for a 3-hop path, as the number of total nodes increases the minimum degree reduces from $N/2$ for $N = 100$ total nodes to about $N/4$ for $N = 1700$ total nodes. For anonymous systems like TOR [24] and Mixminion [25] the current norm is a 3-hop path. If a 4-hop path is used, the minimum degree even when $N = 1700$ is below 100 nodes. In our next simulation we measured the entropy for paths of different hop lengths for a $N = 500$ node network. The results are shown in figure 6.2. We varied the graph degree from $D = 5$ to $D = 100$ and measured the entropy value for each path of

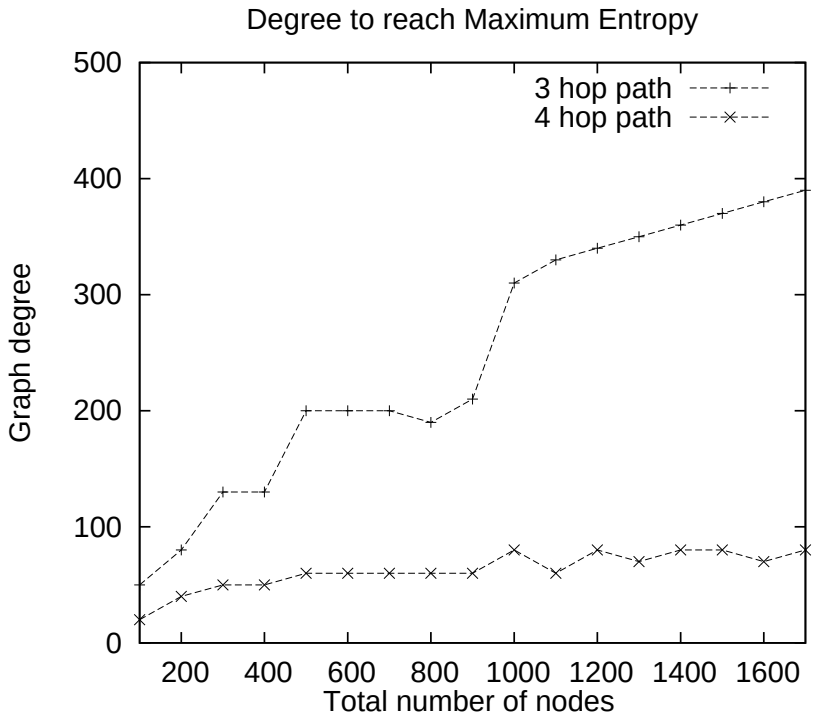

Figure 6.1. Minimum degree to reach maximum entropy for a random expander topology.

different hop length. We found that a 3-hop path reaches close maximum entropy for a degree of 50. A 4-hop path reaches maximum entropy for a degree of only 25.

Figure 6.3 shows the minimum number of hops the path must have reach maximum entropy. We compared fully connected graph with a random expander graph and biased expander graphs with biasing values $SBIAS = 3, 9$, and 15. We found that biasing the expander construction does not lead to a topology that requires significantly more hops to reach maximum entropy. This is important because it allows us to move towards better performance without compromising the security properties of the anonymous network.

In our next simulation we measured the median link RTT of a shaped expander graph with $N = 100$ total nodes with degree $D = 20$. We varied the shaping bias

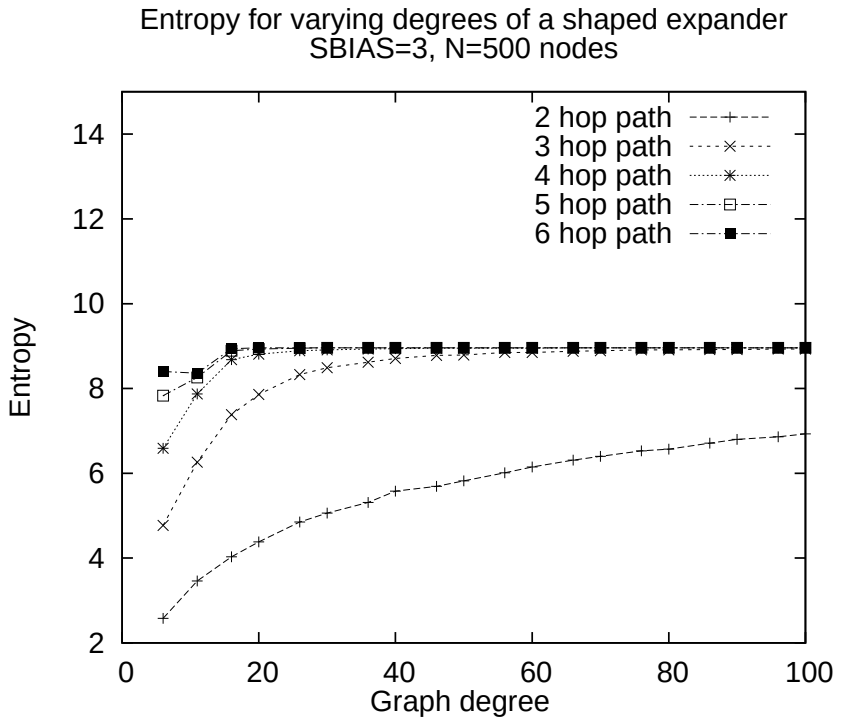

Figure 6.2. Entropy increase with degree. SE=3, N=500, D=20.

of the expander from $SBIAS = -15$ to $SBIAS = 15$. Figure 6.4 plots the median link RTT against varying $SBIAS$. We found that the median link RTT falls from 90 msec to 40 msec as the shaping bias increases.

Figure 6.5 shows the same result for a graph with $N = 500$ nodes and degree $D = 20$. We varied the shaping bias from $SBIAS = 0$ to $SBIAS = 15$. We found that the median link rtt reduces from 55 msec to 36 msec as $SBIAS$ increases from 0 to 9. After $SBIAS = 9$ the distributed construction methodology we used in our simulation repeatedly selects the same links. Hence, we do not see a continued improvement in link RTT as biasing continues to increase after $SBIAS = 9$.

We then measured the median path RTT for a 3-hop path for different shaped expander graphs constructed using varying biasing values. The biasing values used were $SBIAS = 0, 1, 3$, and 9. We also varied the routing bias from $RBIAS = 0$ to

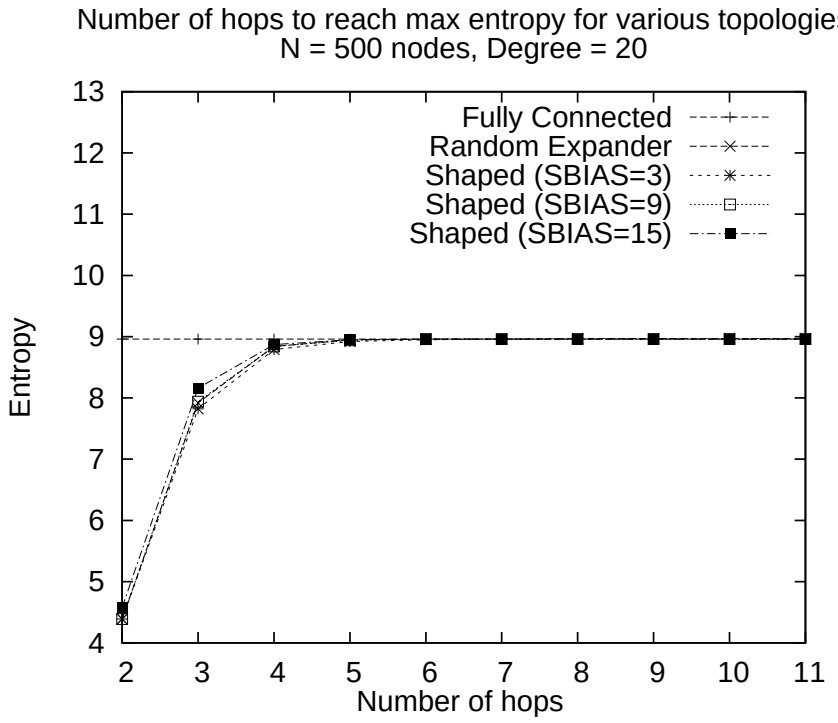

Figure 6.3. Number of hops to reach maximum entropy for different topologies. N=500, D=20.

$RBIAS = 15$. We found that the expander topology with highest bias, $SBIAS = 9$ gave the best performance. Increasing the bias after $SBIAS = 9$ did not significantly improve performance.

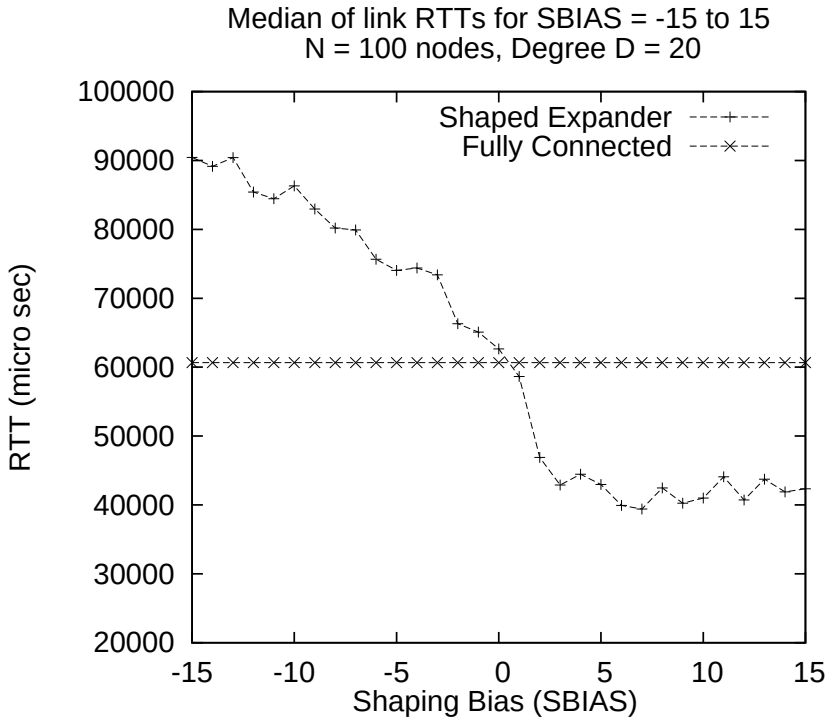

Figure 6.4. Median link RTT for shaping bias -15 to 15. N=100, D=20.

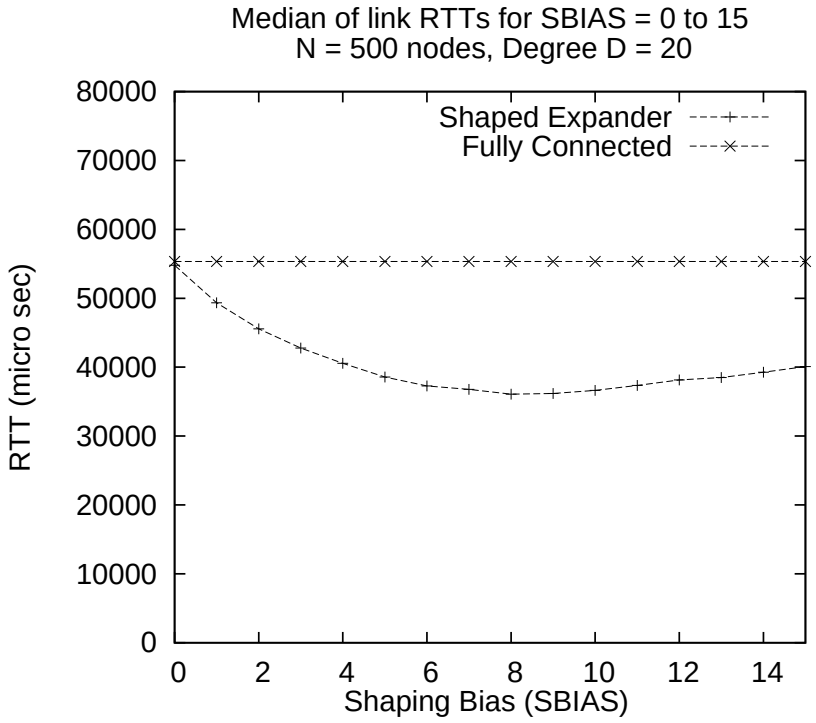

Figure 6.5. Median link RTT for shaping bias 0 to 15. N=500, D=20.

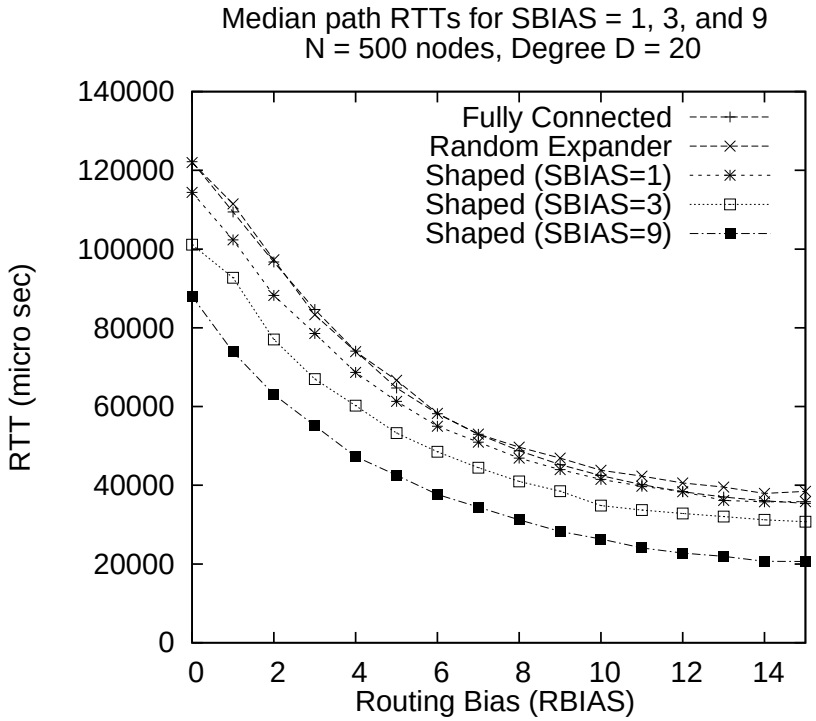

Figure 6.6. Median path RTT for shaping bias = 0,1,3,9. N=500, D=20.

REFERENCES

[1] D. Kesdogan, D. Agarwal, and S. Penz, "Limits of anonymity in open environments," in *Proc. Information Hiding, 5th International Workshop (IH)*, Oct. 2002.

[2] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–88, Feb. 1981.

[3] G. Danezis, "Statistical disclosure attacks: Traffic confirmation in open environments," in *Proc. Security and Privacy in the Age of Uncertainty (SEC)*, May 2003.

[4] C. Troncoso, B. Gierlichs, B. Preneel, and I. Verbauwhede, "Perfect matching statistical disclosure attacks," in *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, N. Borisov and I. Goldberg, Eds. Leuven, Belgium: Springer, July 2008, pp. 2–23.

[5] ——, "Perfect matching statistical disclosure attacks," in *Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008)*, N. Borisov and I. Goldberg, Eds. Leuven, Belgium: Springer, July 2008, pp. 2–23.

[6] G. Danezis, C. Diaz, and C. Troncoso, "Two-sided statistical disclosure attack," in *Proceedings of Privacy Enhancing Technologies, 7th International Workshop, PET 2007*, ser. Lecture Notes in Computer Science, N. Borisov and P. Golle, Eds., vol. 4776. Ottawa,Canada: Springer-Verlag, 2007, pp. 30–44.

[7] D. Kesdogan and L. Pimenidis, "The hitting set attack on anonymity protocols," in *In: Proceedings of 6th Information Hiding Workshop (IH 2004). LNCS.* Springer Verlag, 2004, pp. 326–339.

[8] K. Bennett, C. Grothoff, T. Horozov, I. Patrascu, and T. Stef, "Gnunet – a truly anonymous networking infrastructure," in *Proc. Privacy Enhancing Technologies Workshop (PET)*, Mar. 2002.

[9] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," *Lecture Notes in Computer Science*, vol. 2009, pp. 46–66, 2001.

[10] V. Scarlatta, B. Levine, and C. Shields, "Responder anonymity and anonymous peer-to-peer file sharing," in *Proc. IEEE Intl. Conference on Network Protocols (ICNP)*, Nov. 2001.

[11] R. Sherwood, B. Bhattacharjee, and A. Srinivasan, "P5: A protocol for scalable anonymous communication," in *Proc. 2002 IEEE Sym. on Security and Privacy*, May 2002.

[12] 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.

[13] R. Motwani and P. Raghavan, *Randomized algorithms.* New York, NY, USA: ACM, 1996, vol. 28, no. 1.

[14] N. Mathewson and R. Dingledine, "Practical traffic analysis: Extending and resisting statistical disclosure," in *Proc. Privacy Enhancing Technologies workshop (PET)*, May 2004.

[15] O. Berthold and H. Langos, "Dummy traffic against long-term intersection attacks," in *Proc. Privacy Enhancing Technologies Workshop (PET)*, Apr. 2002.

[16] V. Shmatikov and M.-H. Wang, "Timing analysis in low-latency mix networks: attacks and defenses," in *Proceedings of ESORICS 2006*, 2006, pp. 18–33.

[17] C. Díaz and A. Serjantov, "Generalising mixes," in *Proc. Privacy Enhancing Technologies workshop (PET)*, March 2003.

[18] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/9910332

[19] P. R. Zimmermann, *The official PGP user's guide*. Cambridge, MA, USA: MIT Press, 1995.

[20] S. Dusse, P. Hoffman, B. Ramsdell, L. Lundblade, and L. Repka, "S/mime version 2 message specification," , United States, 1998.

[21] P. Graham, "A plan for spam,"
Available at `http://www.paulgraham.com/spam.html`, Aug. 2002.

[22] T. Meyer and B. Whateley, "Spambayes: Effective open-source, bayesian based, email classification," in *Proc. Conference on Email and Anti-Spam (CEAS)*, Jul. 2004.

[23] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos, "An evaluation of naive bayesian anti-spam filtering," in *Proc. Workshop on Machine Learning in the New Information Age*, May 2000.

[24] P. S. R. Dingledine, N. Mathewson, "Tor: The next-generation onion router," in *Proc. 13th USENIX Security Symposium*, Aug. 2004.

[25] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a type III anonymous remailer protocol," in *Proc. 2003 IEEE Symposium on Security and Privacy*, May 2003.

[26] L. Weinstein, "Spam wars," *Communications of the ACM*, vol. 46, no. 8, p. 136, 2003.

[27] N. Mallesh and M. Wright, "Countering statistical disclosure with receiver-bound cover traffic," in *Proceedings of ESORICS 2007, 12th European Symposium On Research In Computer Security, Dresden, Germany, September 24-26, 2007, Proceedings*, vol. 4734, Sep 2007.

[28] C. Diaz and A. Serjantov, "Generalising mixes," in *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*, R. Dingledine, Ed. Springer-Verlag, LNCS 2760, March 2003, pp. 18–31.

[29] O. Berthold, A. Pfitzmann, and R. Standtke, "The disadvantages of free mix-routes and how to overcome them," in *Proc. Intl. Workshop on Design Issues in Anonymity and Unobservability*, Jul. 2000.

[30] G. Danezis, "Mix-networks with restricted routes," in *Proc. Privacy Enhancing Technologies workshop (PET)*, Mar. 2003.

[31] "King dataset," `http://pdos.csail.mit.edu/p2psim/kingdata/`.

[32] M. Sherr, M. Blaze, and B. T. Loo, "Scalable Link-Based Relay Selection for Anonymous Routing," in *9th Privacy Enhancing Technologies Symposium (PETS '09)*, August 2009.

[33] R. Snader and N. Borisov, "A tune-up for Tor: Improving security and performance in the Tor network," in *Proceedings of the Network and Distributed Security Symposium - NDSS '08*. Internet Society, February 2008.

[34] C. Law and K.-Y. Siu, "Distributed construction of random expander networks," in *In IEEE Infocom*, 2003, pp. 2133–2143.

BIOGRAPHICAL STATEMENT

Nayantara Mallesh received her M.S. and Ph.D. degrees in Computer Science from The University of Texas at Arlington in 2005 and 2010, respectively. Prior to joining UTA she was a member technical staff at Metro-Optix from 2000 to 2003. She was also a software engineer at Cognizant Technology Solution's Telecom Research Center. Her current research interests include anonymous communications, online privacy, pervasive and location privacy, and performance of anonymous communication systems. She is a member of Upsilon Pi Epsilon, a computing honor society and a student member of ACM.