Countering Statistical Disclosure with Receiver-bound Cover Traffic

Nayantara Mallesh and Matthew Wright

Department of Computer Science and Engineering,
The University of Texas at Arlington
{nayantara.mallesh,mwright}@uta.edu
http://isec.uta.edu

Abstract. Anonymous communications provides an important privacy service by keeping passive eavesdroppers from linking communicating parties. However, using long-term statistical analysis of traffic sent to and from such a system, it is possible to link senders with their receivers. Cover traffic is an effective, but somewhat limited, counter strategy against this attack. Earlier work in this area proposes that privacy-sensitive users generate and send cover traffic to the system. However, users are not online all the time and cannot be expected to send consistent levels of cover traffic, drastically reducing the impact of cover traffic. We propose that the mix generate cover traffic that mimics the sending patterns of users in the system. This *receiver-bound cover* helps to make up for users that aren't there, confusing the attacker. We show through simulation how this makes it difficult for an attacker 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 these attacks to succeed. When our approach is used in combination with user-generated cover traffic, the attack takes a very long time to succeed.

Key words: privacy-enhancing technologies, cover traffic, anonymity

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, network partition, and simple user unwillingness to provide so many messages. Additionally, the costs of these messages can cause the system to not scale well with the number of users. In anonymity, this is a substantial matter for security, as the greater the number of users, the larger the crowd into which one can blend [1].

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.

The *statistical disclosure attack* is a particularly effective form of intersection, in which the attacker isolates his attack against a single user, which we will call Alice. The statistics used in this attack are the frequencies with which each recipient gets a message from the system. By taking differences between the frequencies observed when Alice is active and those observed when she is not active, the attacker can estimate Alice's contribution to the recipient set. This attack has been studied previously and is well-understood [3–5].

In this work, we explore defenses against intersection attacks such as statistical disclosure. In particular, we study the relative effectiveness of different defenses, and we present the first in-depth study of the idea of sending cover traffic to recipients that are outside of the system. To date, the possible defenses against intersection attacks have been limited to two basic techniques: the user sending more cover traffic into the network and increasing random delays for messages in the system. We explore the idea of how the presence and cover traffic of other users surprisingly fails to provide any help to the user. We also demonstrate that, with some additional cost, the system can significantly improve its defense against intersection attacks by sending dummies to recipients outside of the system. As this may not be appreciated by all recipients, we discuss ways in which this technique could be made practical.

In the next section, we describe our model and the statistical disclosure attack in more detail. We then motivate the two types of cover traffic that we are studying and analyze their effects in Section 3. Section 4 presents our simulation model and results. Discussion and analysis of the feasibility and costs of the cover traffic methods is presented in Section 6. We discuss related work in Section 7 and then conclude.

2 Statistical Disclosure

The *Statistical Disclosure Attack (SDA)* described by Danezis [4] is a long-term intersection attack against mix-based systems. SDA is an extension of the *disclosure attack* introduced by Kesdogan [3]. In this section, we first explain the network model used, then describe statistical disclosure attacks. We discuss why cover traffic delays statistical disclosure and how it can be used to counter this attack.

2.1 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. The mix

network may consist of a single mix or a network of connected mixes. The attacker is a global adversary who can observe all links from senders to the mix and all links from the mix to recipients. The target of the adversary is the sender Alice and the adversary's aim is to expose the set of recipients with whom Alice communicates.

In each round, b senders, sometimes including Alice, send messages to a set of recipients via the mix. The attacker observes a number of 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 in the network. We abstract away the mix-system details and refer to a single mix or a cascade of mixes as a *mix*.

2.2 Statistical Disclosure

Danezis' SDA is a probability-based approach to the disclosure attack and is a practical way to expose Alice's set of recipients [4]. The attacker makes observations in a number of rounds in which Alice participates and in each round records the recipient set in an observation vector $\overrightarrow{o}$. Each element of $\overrightarrow{o}$ contains the probability that the corresponding recipient has received a message from Alice in that round. The attacker models the behavior of senders other than Alice, known as the *background*, by recording their activity when Alice does not participate. Vector $\overrightarrow{u}$ captures the background model, in which a given element of $\overrightarrow{u}$ is the probability with which background senders send to the corresponding recipient. The attacker records $\overrightarrow{o}$ values over a large number of observations and takes the mean of $\overrightarrow{o}$ as $\overline{O}$. The mean of the background traffic observations is obtained and stored in $\overline{U}$. Alice's likely set of recipients can be determined by solving the below equation for vector $\overrightarrow{v}$:

$$\overline{O} = \frac{\overline{m}.\overrightarrow{v} + (\overline{n} - \overline{m})\overrightarrow{u}}{\overline{n}}$$

Here $\overline{m}$ is the average number of messages sent by Alice in each round and $\overline{n}$ is the average total number of messages sent by all senders, including Alice, in each round. The vector $\overrightarrow{v}$ denotes the sending behavior of Alice. Each element of $\overrightarrow{v}$ is the probability that Alice sends a message to the corresponding recipient in some round. An element of $\overrightarrow{v}$ will have a value of 0 for a recipient who is not in Alice's recipient set and will have a value greater than 0 and less than 1 for a recipient who belongs to Alice's recipient set. $\overrightarrow{v}$ is obtained from the above equation by substituting $\overline{U}$, the mean of background traffic, and $\overline{O}$, the mean of attacker observations in each round. The indices with the highest values in $\overrightarrow{v}$ correspond to the most likely recipients of Alice.

Mathewson and Dingledine extend SDA to pool mixes [5]. Their work relaxes some of the assumptions made in the original work [4]. A pool mix, as described

in [6], operates by dispersing incoming messages from a given round across a number of later rounds. In each round, the mix chooses a set of messages with uniform probability and sends them to their respective recipients. When Alice sends dummy messages along with real messages, it becomes more difficult for the attacker to successfully perform statistical analysis. Dummy messages increase the average number of messages from the sender per round, as seen by the attacker, which substantially affects the results of statistical analysis. The attacker needs more observations to compensate for the presence of dummies and hence it takes a significantly longer time for the attacker to correctly identify Alice's set of recipients.

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 [7]. This means that most senders communicate with a few well-known recipients in addition to other less-known recipients. The well-known recipients hence communicate with many senders and thus receive more messages during their communication lifetime. Background senders tend to send more messages to their more well-known recipients rather than to the lesser-known ones. Alice, however, sends messages equally to all of her recipients.

In reality, most senders are not online all of the time. It is difficult for many users to consistently send cover traffic, as it requires them to be online all the time, without fail. This problem is potentially alleviated when the mix carries the onus of sending cover traffic. In the rest of this paper, we use the model of Mathewson and Dingledine to study the effectiveness of padding generated by the users and by the mix.

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 [5]. 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.

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, 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 2.2 to find $\overrightarrow{v}$ 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 that background dummies do not help against this informed attacker, and that Alice cannot rely on help from her fellow users.

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 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 6.

4 Simulation

Using the basic sender-mix-receiver model described in Section 2, we simulated the process of sending messages 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.

4.1 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 [5].

Attacker Algorithm The attacker algorithm is based on the statistical analysis approach *Attacking pool mixes and mix networks* described in [5]. 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 dummies described in Section 3.

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: In each round, the pool mix receives messages from a number of senders. Alice may or may not participate in a given round. At the end of each round, the mix chooses outgoing messages from the pool with equal

probability. P_{delay} is the probability that a message in the mix pool remains in the pool until it is sent out in a later round. The mix applies P_{delay} to each message in the pool and decides if the message will exit the mix in the current round or not [6]. For our simulations we varied P_{delay} from 0.1 to 0.9. For simulations where P_{delay} does not vary, we set $P_{delay} = 0.1$.

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.

4.2 Cover Traffic Scenarios

The simulations in [5] 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.

Alice and Background Cover Traffic We first study how dummy messages sent by users other than Alice affects statistical analysis. We set $N = 2^{16}$ be the number of senders. Each of the $N - 1$ other senders apart from Alice, called background senders, generate 0 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 that 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 dummy 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.

Receiver-bound Cover Traffic We also evaluate how RB cover traffic originating at the mix impacts statistical analysis. 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. The recipients for the dummy messages are chosen uniformly at random from the set of recipients. For our simulations we used the following dummy generation policy at the mix:

Receiver-bound Cover Policy. We ran simulations with the number of dummy messages per round at 100%, 200%, and 300% additional traffic. In all cases, the mix observes the number of real messages in each round and sends between one to three dummy messages for each real message, according to the amount of RB cover traffic desired. Fractional amounts are possible by sending according to a uniform distribution. We use these fixed values for simplicity; in reality, the mix must choose a random number of RB dummies per round based on a function of the number of real messages exiting the mix in that round.

The recipient of each dummy message is chosen uniformly at random from the set of recipients. This is somewhat unrealistic, as the mixes may not know the full set, but a reasonable approximation can be constructed by using previously observed recipients and a selection of recipient addresses from the general population. Dummy messages travel from the mix to the recipient 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 recipient.

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.

5 Results

In this section we present the results of our simulations. Please note the use of logarithmic scales in our graphs.

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 1 shows that as the attacker tries to expose more number of recipients, the amount of observation rounds significantly increases. In comparison, Figure 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.

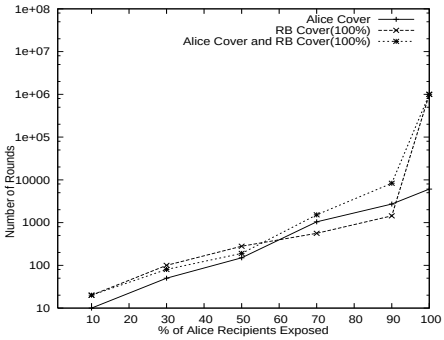

Fig. 1. Median rounds to identify a subset of Alice's recipients. Background **(BG)** volume = 125 messages/round. Mix delay probability P_{delay}=0.5

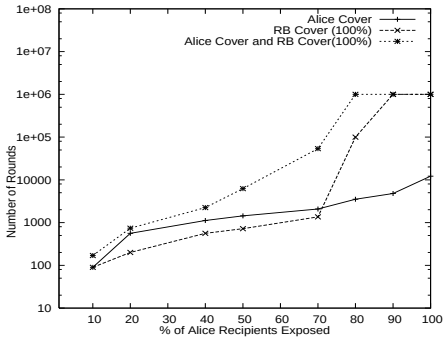

Fig. 2. Median rounds to identify a subset of Alice's recipients. Background **(BG)** volume = 1700 messages/round. Mix delay probability P_{delay}=0.5

5.2 Effect of Background Senders

The graph in Figure 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 generate cover traffic at 10%

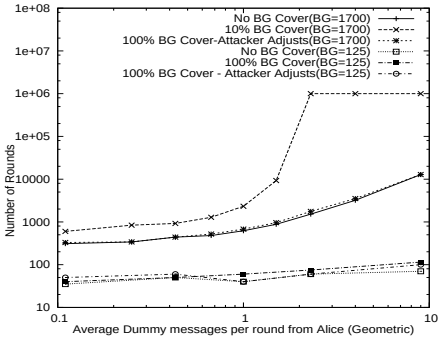

Fig. 3. Effect of Background Cover and Attacker Adjustment. Median rounds to guess 10 recipients.

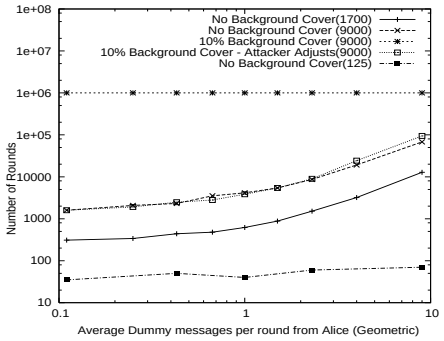

Fig. 4. Effect of increase in Background Volume. Median rounds to guess 10 recipients.

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.

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 were no dummies. Figure 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 amount of background cover traffic is required, rather than the fixed percentage of real traffic that we have studied in this paper.

Larger Number of Participants Figure 4 shows that as the number of participants in the mix increases, the anonymity of individual participants correspondingly increases. In this simulation we increased to 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.

5.3 Effect of Receiver-bound Cover

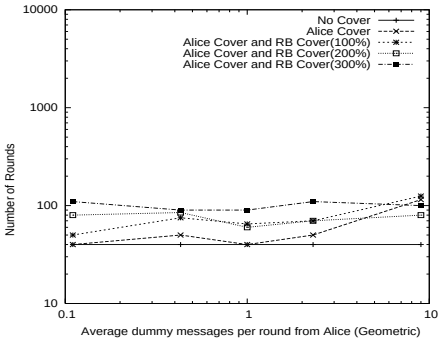

Fig. 5. Effect of RB cover traffic. Median rounds to guess 10 recipients. Background **(BG)** volume = 125 messages/round.

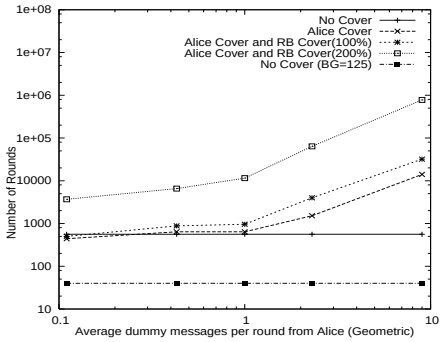

Fig. 6. Effect of RB cover traffic. Median rounds to guess 10 recipients. Background **(BG)** volume = 1700 messages/round.

Figures 5 and 6 show the effect of RB cover traffic. The mix generates RB dummies equal to the number of real messages per round. The number of dummies is shown along the x-axis. 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 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 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.

Figures 7 and 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.

6 Discussion

In Section 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:

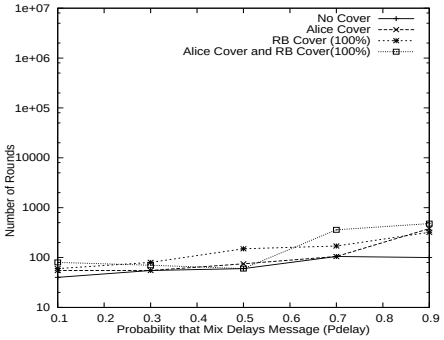

Fig. 7. Effect of increased delay distribution in the mix. Median rounds to guess 10 recipients. Background **(BG)** volume = 125 messages/round.

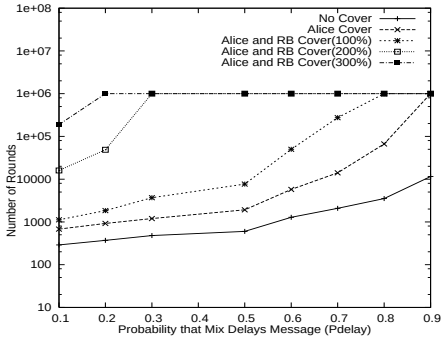

Fig. 8. Effect of increased delay distribution in the mix. Median rounds to guess 10 recipients. Background **(BG)** volume = 1700 messages/round.

- 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.

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.

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 [8] or S-Mime [9] but with random payload bytes (in Radix-64). The receiver could attempt to decrypt the random payload and discard the email when 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 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.

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 [10–12]. 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.

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 [13] 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 [14].

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 [15]. 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.

7 Related Work

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 [16]. Mathewson and Dingledine suggest, and then dismiss, this approach in a footnote of their work on statistical disclosure [5]. 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 [17]. 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 of our work indicate that this approach can also help prevent intersection attacks.

System for anonymous peer-to-peer services, such as GNUnet [18], Freenet [19], and APFS [20], 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[21]. 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 multi-

cast approach would be one way to do receiver-bound cover traffic in mix-based anonymity systems, it would only work in non-encrypted communications.

8 Conclusions and Future Work

Anonymous communications remain challenging in the face of determined and powerful attackers. No matter how secure the process of mixing becomes, inconsistent usage patterns can give the attacker enough information to link users with their communication partners over time. Prior work had developed the notion of statistical disclosure as a powerful form of this attack. In this work, we explored defenses against this attack in greater depth. We found that the cover traffic of other users is surprisingly ineffective in protecting Alice, our user of interest; techniques to hide the amount of real traffic could help. Alice's own cover traffic has a limited effect on its own, or in combination with greater delays in the mix system. We proposed receiver-bound cover traffic and showed that it can have a substantial benefit to the user. We then discussed in detail the implications of using such an approach; we believe that it is feasible, and that the improvement in privacy could well be worth the costs.

Much work remains before receiver-bound cover traffic could be put into place. First, we need to have a deeper study of the use of unencrypted receiver-bound dummies. It is unclear whether it is a pure arms race between defense and attack, or whether one side has a clear advantage. We suggest that the attacker would find that deep content analysis does not scale well, while creating realistic automated messages is a well-understood problem from spam email generation.

9 Acknowledgements

This work is supported in part by a grant from the National Science Foundation under award CNS-0549998. We are grateful to Nick Mathewson for his statistical disclosure simulation code and to Thomas Heydt-Benjamin and Banessa Defend for useful discussions. We thank the reviewers for their detailed comments and suggestions which greatly helped to improve the clarity of this work.

References

1. Acquisti, A., Dingledine, R., Syverson, P.: On the economics of anonymity. In: Proc. Financial Cryptography (FC). (Jan. 2003)
2. Chaum, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM **24**(2) (Feb. 1981) 84–88
3. Kesdogan, D., Agarwal, D., Penz, S.: Limits of anonymity in open environments. In: Proc. Information Hiding, 5th International Workshop (IH). (Oct. 2002)
4. Danezis, G.: Statistical disclosure attacks: Traffic confirmation in open environments. In: Proc. Security and Privacy in the Age of Uncertainty (SEC). (May 2003)

5. Mathewson, N., Dingledine, R.: Practical traffic analysis: Extending and resisting statistical disclosure. In: Proc. Privacy Enhancing Technologies workshop (PET). (May 2004)
6. Díaz, C., Serjantov, A.: Generalising mixes. In: Proc. Privacy Enhancing Technologies workshop (PET). (March 2003)
7. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286** (1999) 509–512
8. Zimmermann, P.R.: The official PGP user's guide. MIT Press, Cambridge, MA, USA (1995)
9. Dusse, S., Hoffman, P., Ramsdell, B., Lundblade, L., Repka, L.: S/mime version 2 message specification (1998)
10. Graham, P.: A plan for spam.
Available at `http://www.paulgraham.com/spam.html` (Aug. 2002)
11. Meyer, T., Whateley, B.: Spambayes: Effective open-source, bayesian based, email classification. In: Proc. Conference on Email and Anti-Spam (CEAS). (Jul. 2004)
12. Androutsopoulos, I., Koutsias, J., Chandrinos, K., Paliouras, G., Spyropoulos, C.: An evaluation of naive bayesian anti-spam filtering. In: Proc. Workshop on Machine Learning in the New Information Age. (May 2000)
13. R. Dingledine, N. Mathewson, P.S.: Tor: The next-generation onion router. In: Proc. 13th USENIX Security Symposium. (Aug. 2004)
14. Danezis, G., Dingledine, R., Mathewson, N.: Mixminion: Design of a type III anonymous remailer protocol. In: Proc. 2003 IEEE Symposium on Security and Privacy. (May 2003)
15. Weinstein, L.: Spam wars. Communications of the ACM **46**(8) (2003) 136
16. Berthold, O., Langos, H.: Dummy traffic against long-term intersection attacks. In: Proc. Privacy Enhancing Technologies Workshop (PET). (Apr. 2002)
17. Shmatikov, V., Wang, M.H.: Timing analysis in low-latency mix networks: attacks and defenses. In: Proceedings of ESORICS 2006. (2006) 18–33
18. Bennett, K., Grothoff, C., Horozov, T., Patrascu, I., Stef, T.: Gnunet – a truly anonymous networking infrastructure. In: Proc. Privacy Enhancing Technologies Workshop (PET). (Mar. 2002)
19. Clarke, I., Sandberg, O., Wiley, B., Hong, T.W.: Freenet: A distributed anonymous information storage and retrieval system. Lecture Notes in Computer Science **2009** (2001) 46–66
20. Scarlatta, V., Levine, B., Shields, C.: Responder anonymity and anonymous peer-to-peer file sharing. In: Proc. IEEE Intl. Conference on Network Protocols (ICNP). (Nov. 2001)
21. Sherwood, R., Bhattacharjee, B., Srinivasan, A.: P5: A protocol for scalable anonymous communication. In: Proc. 2002 IEEE Sym. on Security and Privacy. (May 2002)