The Reverse Statistical Disclosure Attack

Nayantara Mallesh and Matthew Wright

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

Abstract. 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 paper, 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.

1 Introduction

Mix-based anonymity systems [1] 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 [2] 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.

1.1 Contributions

In this paper, 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) (§4). 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. Additionaly 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 (§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) [3] and the Perfect Matching Disclosure Attack (PMDA) [4]. We discuss these in more detail in §3, 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) to study RSDA using different mix network configurations. Cover traffic has been recognized as an effective way to counter Statistical Disclosure Attacks [5, 6]. Hence, we also study the effectiveness of cover traffic, including *background cover* and *receiver-bound cover*, as a countermeasure. Our results (§6) 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.

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.

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 [1] 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 [7], 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.

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 [5, 3, 4], 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 [6] for details on how RBC is used to counter SDA.

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.

3 Statistical Disclosure Attacks

In this section, we describe the Statistical Disclosure Attack, which is central to the function of RSDA and which we use as a basis for comparison. We also describe the Two-Sided Statistical Disclosure Attack and the Perfect Matching Disclosure Attack, both of which use observations about other senders to inform their method. RSDA uses these observations in a very different way from these existing techniques.

3.1 The (Original) Statistical Disclosure Attack

The Statistical Disclosure Attack (SDA) is a statistical technique for finding the receivers of a single targeted user Alice based on observed inputs to and output from the mix network. The attacker makes observations in a number of *rounds*, i.e. periods during which Alice participates. In each round of observation, the attacker records three pieces of information: n_A, the number of messages sent by Alice; n_B the number of messages sent by senders other than Alice; and $\overrightarrow{o}$ the distribution of messages received by receivers in that round. The attacker records 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 distribution of messages from background senders to receivers in each round in which Alice is not present. The attacker sums $\overrightarrow{u}$ 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. Thus, $\overline{O}$ represents the combined sending behavior of both Alice and the background during the observed rounds, and this can be written as:

$$\overline{O} = \overline{n_A}.D_A + \overline{n_B}.D_N \tag{1}$$

Here $\overline{n_A}$ and $\overline{n_B}$ 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. For receiver i, who is Alice's contact, $0 < D_A[i] < 1$, and for receiver j who is not Alice's contact, $D_A[j] = 0$. D_N represents the sending behavior of background senders and is obtained by observing rounds in which Alice does not participate i.e. $D_N = \overline{U}/\overline{n_B}$. If the attacker is unable to collect background statistics before Alice begins communicating, D_N can be approximated as $D_N[i] = \frac{1}{N}\ \forall\ i$, 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 (1) and picking m receivers with the highest $D_A[i]$ values.

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 [5]. 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.

3.2 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) [3] 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.

As we note in Section 1, 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.

3.3 Perfect Matching Disclosure Attack

In the Perfect Matching Disclosure Attack (PMDA) [4], 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.

4 Reverse Statistical Disclosure Attack

In the Reverse Statistical Disclosure Attack (RSDA), the attacker first applies the SDA (as described in Section 3.1) 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.

Forward Observation In each round of observation the attacker records information in the forward direction as described in Section 3.1. This allows the attacker to calculate D_A, a set of scores representing Alice's estimated sending behavior.

Reverse Observation For reverse observation, the attacker also applies SDA to all the other users. For a given 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 attacker must also observe or estimate D_N^X, the distribution of all the users' sending behavior without X. Based on these observations, the attacker can use the following equation:

$$\overrightarrow{O} = \overrightarrow{n_X}.D_X + \overrightarrow{n_B}.D_N^X \tag{2}$$

With these observations, the attacker can apply Eqn. 2 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.

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:

$$\widehat{D}_A = \frac{v_f.D_A + v_r.D_R}{v_f + v_r} \tag{3}$$

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 Simulation Setup

We simulated the process of sending and receiving messages via a mix network according to the model described in Section 2. The parameters used in our simulations are discussed in this section and summarized in Table 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.

Parameter	Value	Description
N	100	Number of users in the system
m	10	Number of Alice's contacts
B	200	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
$RBCVOL$	100%	RBC volume as per cent of real messages/round
$CUTOFF$	10^5 10^6	Simulation cutoff, Threshold Mix Simulation cutoff, Binomial Mix

Table 1. Simulation parameter values

5.1 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 $P_{delay} = 0.2$.

5.2 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.

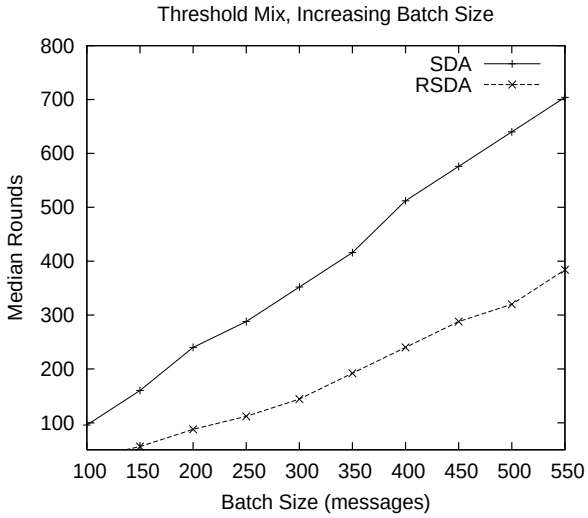

Fig. 1. Median rounds to identify a 50% of Alice's recipients. Threshold mix with no cover traffic

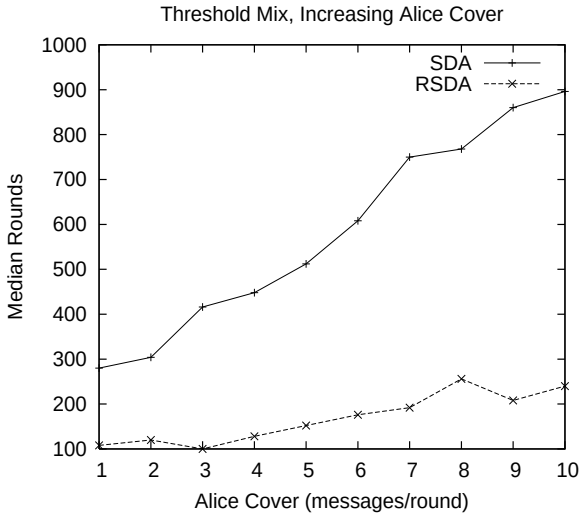

Fig. 2. Median rounds to identify a 50% of Alice's recipients with Alice Cover. Threshold mix with $B = 200$

5.3 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 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 [6] 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.

6 Results

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

6.1 Simple Threshold Mix

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 1. We see that when a threshold mix is used, RSDA outperforms SDA especially for higher batch sizes.

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 increasing from 1.0 to 10.0 messages/round. Figure 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.

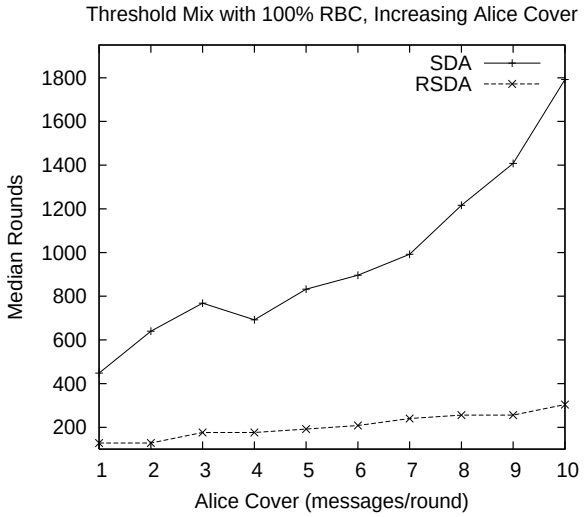

Fig. 3. Median rounds to identify 50% Alice's recipients. Threshold mix with $B = 200$, $RBCVOL = 100\%$

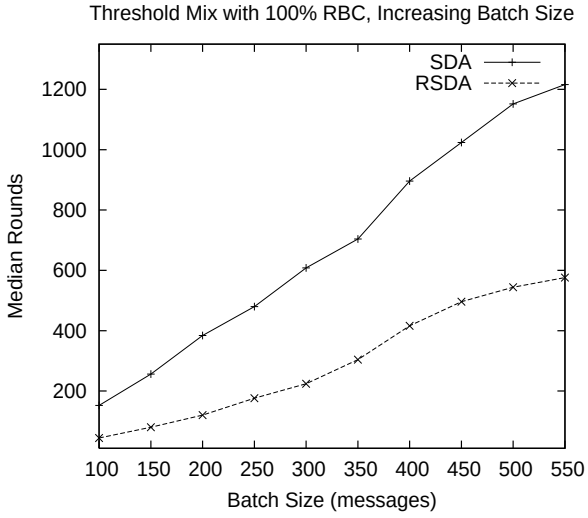

Fig. 4. Median rounds to identify 50% Alice's recipients. Threshold mix, $RBCVOL = 100\%$

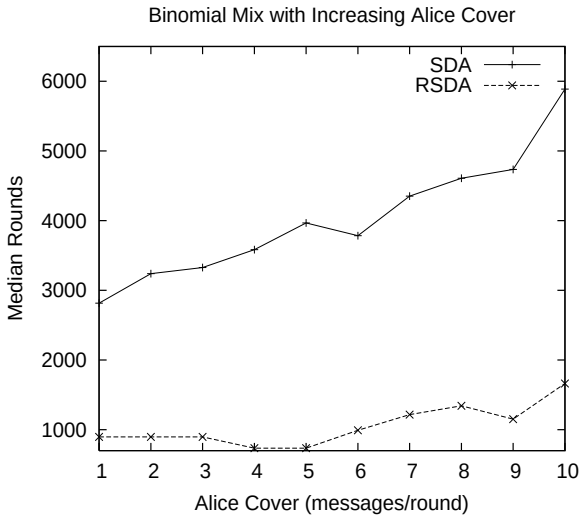

Fig. 5. Median rounds to identify 50% Alice's recipients. Binomial mix with increasing Alice cover

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 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 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 4. SDA and RSDA are compared for increasing values of mix batch size.

6.2 Binomial Mix

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. We see that, like in the threshold case, RSDA is able to outperform SDA.

Alice and Receiver-bound Cover In this simulation we show the 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 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.

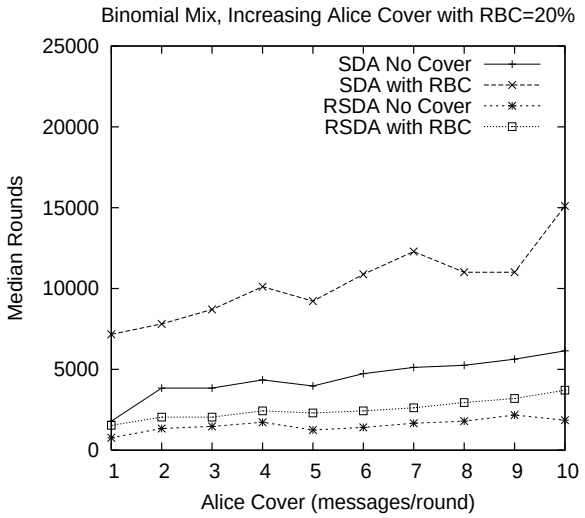

Fig. 6. Median rounds to identify 50% Alice's recipients. Binomial mix with RBC=20%

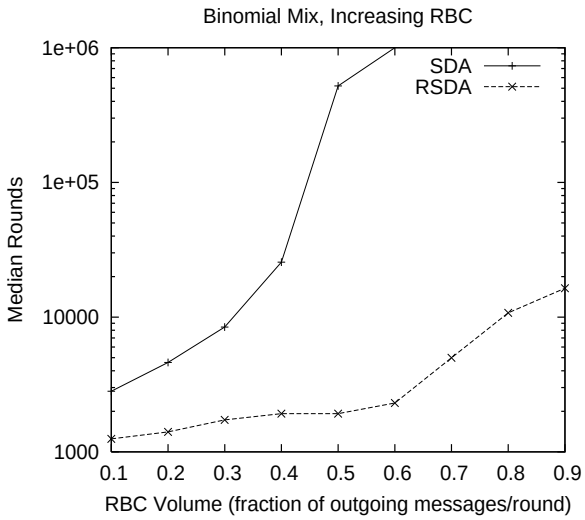

Fig. 7. Median rounds to identify 50% Alice's recipients. Binomial mix with increasing RBC

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 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.

7 Conclusions

In this paper, we have described and evaluated RSDA. RSDA is based on the assumption that the attacker is interested in all of Alice's contacts, not just the people to whom she initiates messages. We believe that this is a more realistic representation of the attacker's goals. In this model, we showed that taking other users' sending behavior into account, the attacker could better identify Alice's contacts than when just using SDA.

References

1. Chaum, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM **24**(2) (Feb. 1981) 84–88
2. Danezis, G.: Statistical disclosure attacks: Traffic confirmation in open environments. In: Proc. Security and Privacy in the Age of Uncertainty (SEC). (May 2003)
3. Danezis, G., Diaz, C., Troncoso, C.: Two-sided statistical disclosure attack. In Borisov, N., Golle, P., eds.: Proceedings of Privacy Enhancing Technologies, 7th International Workshop, PET 2007. Volume 4776 of Lecture Notes in Computer Science., Ottawa,Canada, Springer-Verlag (2007) 30–44
4. Troncoso, C., Gierlichs, B., Preneel, B., Verbauwhede, I.: Perfect matching statistical disclosure attacks. In Borisov, N., Goldberg, I., eds.: Proceedings of the Eighth International Symposium on Privacy Enhancing Technologies (PETS 2008), Leuven, Belgium, Springer (July 2008) 2–23
5. Mathewson, N., Dingledine, R.: Practical traffic analysis: Extending and resisting statistical disclosure. In: Proc. Privacy Enhancing Technologies workshop (PET). (May 2004)
6. Mallesh, N., Wright, M.: 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. Volume 4734. (Sep 2007)
7. Diaz, C., Serjantov, A.: Generalising mixes. In Dingledine, R., ed.: Proceedings of Privacy Enhancing Technologies workshop (PET 2003), Springer-Verlag, LNCS 2760 (March 2003) 18–31