

On the Sender Cover Traffic Countermeasure against an Improved Statistical Disclosure Attack*

Rajiv Bagai, Huabo Lu and Bin Tang
Department of Electrical Engineering and Computer Science
Wichita State University
Wichita, KS 67260-0083, USA
rajiv.bagai@wichita.edu, hxlu@wichita.edu, bin.tang@wichita.edu

Abstract—The Statistical Disclosure Attack against a particular user of an anonymity system is known to be very effective in determining, after long-term observation of the system, the set of receivers that user sends messages to. This paper first presents an improvement over this attack that, by employing a weighted mean of the observed relative receiver popularity, is more accurate than the original one based upon arithmetic mean. Second, a mathematical analysis is presented of this attack on a model, in which senders blend dummy messages with real ones. It is shown that despite such sender-generated dummy cover traffic, the attack can proceed almost unhindered. The analysis substantiates earlier empirical indications of the ineffectiveness of this countermeasure.

Keywords—anonymity; statistical disclosure; cover traffic

I. INTRODUCTION

Many applications for which the Internet is used today were completely unforeseen back when its underlying architecture was designed and put into practice. In particular, the need for anonymous communication was not anticipated and for such communication there is little support, if any, in its basic framework. Anonymous web surfing, chatting, emailing and evoting are just some examples of applications that require anonymity. Systems that enable anonymous communication for such applications are therefore constructed on top of the existing architecture.

A popular technique to implement an anonymity system is as a *mix* network, as proposed by Chaum [3], which is a collection of proxy nodes that relay messages between *senders* and, possibly overlapping, *receivers* connected to the mix. These intermediate proxies are the fundamental source of the anonymity achieved by such systems.

Several attacks by adversaries on mix-based anonymity systems, along with possible countermeasures, have been studied. Back, Möller and Stiglic [1], and Raymond [10] contain detailed lists of attacks. Of these, the class of long-term *intersection attacks* is one of the strongest. In these attacks, a passive global adversary correlates senders with receivers that they often send messages to, by observing over a long period messages that enter and leave the mix.

The *Statistical Disclosure Attack (SDA)*, proposed by Danezis [4], is a member of this class of attacks that is directed against a single sender. The attack aims to uncover the receivers related to that sender by keeping track of, among others, the observed relative popularity of the various receivers of the system. Mathewson and Dingledine [9] gave an extended version of this attack by removing some of the restrictions on the number of messages flowing through the mix in the original version of [4].

The first contribution of this paper is an improvement on the extended SDA of [9]. Our attack is based upon the *weighted* mean of the observed relative popularity of the receivers over time. This results in a more accurate conclusion than one obtained by the *arithmetic* mean method of [9].

One of the many strategies studied in the past to counter SDA is for senders of the system to send dummy messages along with the real ones. The dummy messages in this strategy are detected and blocked by the mix. Such messages were expected to confuse the frequency tracking of the attack, thereby thwarting it. However, Mallesh and Wright [8] gave empirical results to the contrary. Their experiments indicated that the attack succeeds despite the presence of dummy messages emitted by senders, although their work lacks a mathematical rationale behind the ineffectiveness of this seemingly adequate countermeasure.

The second contribution of this paper is a mathematical analysis of SDA in the mix-based anonymity system model containing sender-generated dummy messages. Our analysis substantiates the experimental findings of [8] by establishing that SDA is not significantly affected by such dummy messages.

The rest of this paper is organized as follows. Section II contains a quick overview of the extended SDA of [9] on the basic anonymous network model containing only real messages. It describes how a global adversary, by observing the system over time, can determine the set of receivers a particular sender usually sends messages to. Section III presents our strengthening of this attack that incorporates the weighted mean of receiver popularity. We show by an example how this results in an attack that is more accurate

* The research described in this paper was partially supported by the United States Navy Engineering Logistics Office contract no. N41756-08-C-3077.

than that of [9], which employs only arithmetic mean. Section IV first extends the basic model to one in which senders may blend dummy messages with real ones. It then shows how, despite such cover traffic, the attack can still be carried out almost unhindered. This mathematical analysis supports the empirical results of [8]. Section V concludes our main results and presents some directions for future work.

II. STATISTICAL DISCLOSURE ATTACK

In this section we give a brief overview of SDA, first proposed by Danezis [4], and later extended by Mathewson and Dingledine [9]. The attack is based upon the disclosure attack of Kesdogan, Agrawal and Penz [7] and has been well studied, as in Mallesh and Wright [8].

The underlying model is that of a mix network, as introduced by Chaum [3], that is connected to some *senders* and, possibly overlapping, *receivers*. The main task of the mix network is to transmit messages sent by any of the senders to their destined receivers, so that they arrive at their destination anonymously, i.e. without any sender identification. Immediate transmission of messages is vulnerable to timing analysis by a global observer that can successfully defeat the intended anonymity. The mix therefore collects a certain number of messages in each *round*, and transmits them simultaneously. Such rounds are repeated *ad infinitum*.

SDA is targeted against a particular sender, called *Alice*. The aim of the attack is to uncover, over a period of time, the subset of receivers that Alice sends messages to, called *Alice's friends*. This is achieved by observing the messages entering and leaving the mix at each round. All other senders are called *background* senders.

Let a_k be the number of messages sent by Alice in round k , and b_k be the number of messages sent by all other senders in that round. Thus, $a_k + b_k$ messages enter and leave the mix in that round. Figure 1 shows the message flow via the mix in Round k . The simple link in the figure from Alice to the mix represents a single connection. The bold links from the background senders to the mix and from the mix to the receivers represent a group of connections.

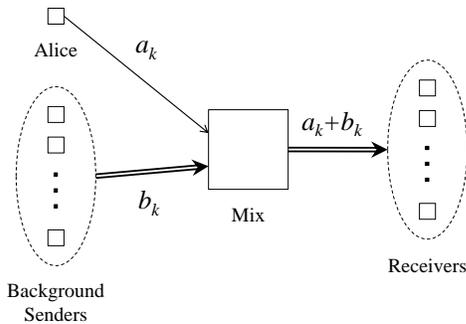


Figure 1. Message flow in Round k

Let the vector \vec{r}_k contain the number of messages arriving at each receiver in that round. SDA employs another vector

\vec{o}_k containing, in a sense, the *observed relative popularity* of each receiver in round k , i.e. the fraction of the total outgoing messages received by each receiver. This vector is defined as

$$\vec{o}_k[i] = \frac{r_k[i]}{a_k + b_k},$$

for any receiver i . It is worth noting that o_k can be obtained easily by observing just the messages leaving the mix.

The vector \vec{O} captures the cumulative observed receiver popularity so far by maintaining a running average of all the previous o_k vectors, i.e. after t rounds, \vec{O} is the average of the vectors in the following set:

$$\{\vec{o}_k \mid 1 \leq k \leq t\}.$$

Alternatively,

$$\vec{O}[i] = \frac{\sum_{k=1}^t \vec{o}_k[i]}{t},$$

for any receiver i .

SDA exploits the fact that for large values of t , the above *actual* receiver popularity approximates the *expected* one, formulated below.

Let the vector \vec{u} denote the observed receiver popularity for messages sent only by the background senders. This can be obtained in much the same way as \vec{O} , with the exception that the \vec{o}_k vectors for only the rounds in which Alice does *not* participate are averaged. In other words, \vec{u} is the average of the vectors in the following set:

$$\{\vec{o}_k \mid 1 \leq k \leq t, \text{ and } a_k = 0\}.$$

The underlying assumption of there being enough rounds in which Alice does not participate, thereby facilitating the computation of \vec{u} , is not an unreasonable one. Most senders connected to an anonymous system, such as users who browse the web, are on-line only some of the time and off-line most of the time. If Alice is an ordinary user, \vec{u} can be obtained easily during rounds that she is off-line.

We also let \bar{m} be the average number of messages sent by Alice in each round, i.e. $\bar{m} = (\sum_{k=1}^t a_k)/t$. Similarly, let $\bar{n} = (\sum_{k=1}^t a_k + b_k)/t$ be the average number of total messages sent in each round.

The goal of SDA is to determine another vector \vec{v} that contains the *relative degrees of friendship with Alice* of all receivers. This vector is similar to \vec{u} , with the exception that it is for messages sent only by Alice. The expected receiver popularity can now be expressed as:

$$\frac{\bar{m}\vec{v} + (\bar{n} - \bar{m})\vec{u}}{\bar{n}}.$$

The above expression is based upon the fact that of the \bar{n} average number of messages sent in a round, \bar{m} messages from Alice should reach receivers according to their degrees of friendship with Alice in \vec{v} , and the remaining $\bar{n} - \bar{m}$ messages from background senders should reach them according

to their degrees of friendship, with all background senders as a whole, in \vec{u} .

As stated earlier, when t is large enough, the above expected popularity approximates \vec{O} , the observed one, i.e.

$$\vec{O} \approx \frac{\bar{m}\vec{v} + (\bar{n} - \bar{m})\vec{u}}{\bar{n}}.$$

By rearranging, we get

$$\vec{v} \approx \frac{\bar{n}\vec{O} - (\bar{n} - \bar{m})\vec{u}}{\bar{m}}. \quad (1)$$

All values on the right side of Equation 1 can be obtained by observing the mix over time, thus making possible a reasonable estimate of the receivers' degrees of friendship with Alice.

III. AN IMPROVED SDA

In the basic SDA presented by Danezis [4], the mix outputs a fixed number of messages in each round, exactly one of which is sent by Alice. Computing \vec{O} as an arithmetic mean of the \vec{o}_k vectors is all that is needed for that model. Also, in that model \vec{u} corresponds to uniform distribution over all receivers, which is fixed and does not need to be computed.

The model presented in Section II is an extension developed by Mathewson and Dingleline [9], in that the number of messages transmitted by the mix in each round can vary, Alice is allowed to send any number of messages in each round, and \vec{u} need not be uniform. However, the SDA of [9] continues to employ the same arithmetic mean method for computing \vec{O} (and \vec{u}), which can be made more accurate by instead employing a weighted mean based upon the total number of messages output by the mix.

As an example, suppose A and B are the only receivers in the system. If in Round 1, A receives 1 message and B receives 3 messages, then $\vec{o}_1 = \langle 0.25, 0.75 \rangle$. Now, if in Round 2, A receives 300 messages and B receives 100 messages, then $\vec{o}_2 = \langle 0.75, 0.25 \rangle$. An arithmetic mean of these vectors gives

$$\vec{O} = \langle 0.5, 0.5 \rangle.$$

On the other hand, a mean weighted by the total number of messages in each round would result in

$$\begin{aligned} \vec{O} &= \left\langle \frac{4(0.25) + 400(0.75)}{4 + 400}, \frac{4(0.75) + 400(0.25)}{4 + 400} \right\rangle \\ &\approx \langle 0.745, 0.255 \rangle, \end{aligned}$$

which better reflects the portion of the *total* number of messages received by the two receivers so far. As the intuition behind \vec{O} is the *cumulative* observed relative popularity of receivers so far, its computation based upon weighted averages is more in line with that intuition.

We thus propose the following definition of \vec{O} :

$$\vec{O}[i] = \frac{\sum_{k=1}^t (a_k + b_k) \vec{o}_k[i]}{\sum_{k=1}^t (a_k + b_k)},$$

which can be simplified to

$$\vec{O}[i] = \frac{\sum_{k=1}^t r_k^{\vec{u}}[i]}{\sum_{k=1}^t (a_k + b_k)},$$

for any receiver i .

The vector \vec{u} should be similarly computed as a weighted average of the \vec{o}_k vectors for rounds in which Alice does not participate.

In order to better study the effect of this change of computation method for \vec{O} and \vec{u} on the effectiveness of SDA to estimate Alice's friends, let us extend this example to a total of 7 rounds, as shown in Table I.

Round Number	Alice to A	Alice to B	Background to A	Background to B
1	0	1	1	2
2	200	0	100	100
3	4	2	104	105
4	80	21	1172	1160
5	0	0	1000	992
6	202	70	1080	1090
7	2	6	12	12
Total	488	100	3469	3461

Table I
MESSAGES SENT TO RECEIVERS A AND B

The above table shows the number of messages sent by Alice and the background senders to the two receivers, A and B , in each of the 7 rounds. While such detailed information is not available to the attacker, we use it to compare the effectiveness of the attack according to the old and new definitions of \vec{O} .

The values \bar{m} and \bar{n} can be determined from Table I to be 84 and 1074, respectively. From Round 5, in which Alice does not send any message, \vec{u} is estimated to be about $\langle 0.502, 0.498 \rangle$. By using these values of \bar{m} , \bar{n} , and \vec{u} in Equation 1, along with the value of \vec{O} as the arithmetic mean of the \vec{o}_k vectors, we get

$$\vec{v} \approx \langle 0.44, 0.56 \rangle.$$

The above value of \vec{v} is misleading as it suggests B being more of Alice's friend than A is. On the other hand, our new definition of \vec{O} as a weighted mean results in

$$\vec{v} \approx \langle 0.81, 0.19 \rangle,$$

which is much closer to its actual value from these 7 rounds of $\langle \frac{488}{488+100}, \frac{100}{488+100} \rangle$, which is about $\langle 0.83, 0.17 \rangle$.

IV. SENDER COVER TRAFFIC

SDA is known to be a very powerful attack that, given time, quite accurately accomplishes its goal. Many defense strategies have been proposed for thwarting SDA. Most of these strategies either (1) introduce some delay for messages within the mix, or (2) introduce some dummy messages that appear to the attacker just like real messages.

The effectiveness of the message delaying countermeasure is limited. Kesdogan, Egner and Büschkes proposed stop-and-go mixes in [6] that hold any incoming message within the mix according to its sender-specified acceptable message latency. Batching strategies of Serjantov, Dingledine and Syverson [11] also spread out exit times for messages. Pool mixes of Díaz and Serjantov [5] incorporate a distribution function for tailoring the anonymity/delay tradeoff that adapts to traffic load fluctuations. Mathewson and Dingledine [9] employ these pool mixes specifically for SDA and study their effect. As all values in Equation 1 are averages computed over the long-term, strategies that introduce a delay of a few rounds within the mix are not particularly effective against SDA. Even when such a technique manages to be somewhat useful for countering SDA, it does so at the expense of extra latency, thus becoming inapplicable to situations that have low latency requirements, such as web browsing or online chatting.

The strategy of injecting dummy messages into the system is relatively more effective. Berthold and Langos present in [2] a method for sending dummy messages on Alice's behalf when she is off-line, but Mathewson and Dingledine [9] mention many problems with that approach. Shmatikov and Wang [12] propose another method in which senders generate dummy messages in advance and send them to the mix, and these messages are used by the mix later, when needed. Although this method is for low-latency networks, it is not suitable for long-term intersection attacks, such as SDA. Malleh and Wright [8] study the effects of sender-generated as well as mix-generated dummy messages on SDA. They show by simulation results that sender-generated dummy traffic is not effective against SDA, while mix-generated is quite effective. We now substantiate their simulation results for the sender-generated case by presenting a mathematical argument for the ease with which SDA can be carried out in this case.

We consider the model where all senders (including Alice) send dummy messages, called *cover traffic*, to the mix. As an assumption of SDA is that message contents are not visible to external observers, to such an observer, these dummy messages are indistinguishable from the real ones. However, the mix is able to tell the dummy messages from the real ones and block them; it transmits only the real messages to their receivers. This assumption of the mix being able to identify dummy messages, whereas an external observer cannot, is not an unreasonable one. Often, messages are

encrypted, and a dummy indicator embedded in an encrypted message can be made to become visible to the mix only after it decrypts that message.

In order to determine the effectiveness of sender cover traffic against SDA, we study the effect of such traffic on the computation of all values in the right side of Equation 1.

The computations of the \vec{o}_k vectors, thus of \vec{O} as well, stay unchanged from before as the messages coming out of the mix are the same as without any cover traffic. For the same reason, \bar{n} is still computed just as before. Computation of the other values in the right side of Equation 1, namely \vec{u} and \bar{m} , are affected somewhat, as analyzed below.

Recall that a_k and b_k are the number of real messages sent by Alice and all other senders, respectively, in Round k . We now let a'_k and b'_k be the respective number of dummy messages sent in that round. An attacker can thus observe $(a_k + a'_k)$ messages being sent by Alice, $(b_k + b'_k)$ messages being sent by other senders, and still $(a_k + b_k)$ messages coming out of the mix. Figure 2 shows the message flow that includes dummy messages via the mix in Round k .

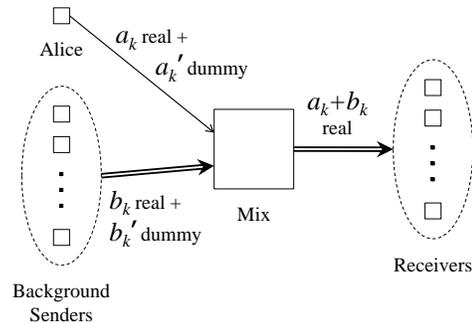


Figure 2. Message flow with dummy messages in Round k

While the computation of the \vec{u} vector still only needs $a_k = 0$, the indistinguishability of a_k and a'_k necessitates its computation during the rounds in which $a'_k = 0$ as well. Therefore, in rounds where Alice has no real message to send, it is in its interest to send some dummy messages nonetheless in order to make those rounds unsuitable for the computation of \vec{u} . If so, \vec{u} can be computed effectively only during rounds when Alice is completely off-line, thereby delaying SDA somewhat.

The value \bar{m} of Equation 1 still needs to be the average number of *real* messages sent per round by Alice. In the presence of indistinguishable dummy messages, this seems to be difficult to determine, especially if the dummy-to-real message volume ratios of senders vary from one round to another. We begin by assuming that these ratios for Alice and the background senders stay the same over all rounds, i.e.

$$\alpha = \frac{a'_{k_1}}{a_{k_1}} = \frac{a'_{k_2}}{a_{k_2}}, \text{ and}$$

$$\beta = \frac{b'_{k_1}}{b_{k_1}} = \frac{b'_{k_2}}{b_{k_2}}, \text{ for all } k_1 \text{ and } k_2.$$

Under this assumption that α and β do not change from one round to another, β is determined easily during any round when Alice is off-line. That in turn leads to an easy determination of α in a round when Alice is on-line and sends messages, as illustrated by the example below. Once α and β are known to the attacker, computing \bar{m} is straightforward.

As an example, suppose in some Round j , Alice is off-line, 100 messages enter the mix, of which 80 exit, i.e.

$$\begin{aligned} b_j + b'_j &= 100, \text{ and} \\ b_j &= 80. \end{aligned}$$

Thus, $b'_j = 20$ and $\beta = 20/80 = 0.25$. Now, if in another Round k , Alice is online, the mix receives 60 messages from Alice, 500 messages from the background senders, and outputs 450 messages, then

$$\begin{aligned} a_k + a'_k &= 60, \\ b_k + b'_k &= 500, \text{ and} \\ a_k + b_k &= 450. \end{aligned}$$

Given that $\beta = 0.25$ is the same in Rounds j as well as k , the above equations can be solved to obtain $a'_k = 10$ and $a_k = 50$, i.e. $\alpha = 10/50 = 0.2$. In other words, 1/6 of total messages sent by Alice in any round are dummy. \bar{m} is therefore 5/6 of the average number of total messages sent by Alice in any round.

The above requirement of the constancy of α and β over all rounds is counterproductive for the anonymity system as, by making the system predictable, it can only assist in the carrying out of the SDA. This requirement also goes against the recommendation stated earlier for Alice to send dummy messages even in rounds where it has no real messages to send.

In a more realistic setting, when the proportion of dummy messages can vary over rounds, \bar{m} is at best approximated. First, an average value of β can be obtained by observing the system over sufficient rounds in which Alice is off-line. That value can then be used to guess a_k for any round in which Alice participates. Since \bar{m} is the average of these guessed a_k values, any inaccuracies in these values likely cancel out over the long-term, resulting in a fairly accurate \bar{m} .

With all four values in the right side of Equation 1, namely \vec{O} , \vec{u} , \bar{m} , and \bar{n} , still fairly easily computable in the presence of dummy messages from senders, it is evident that blending sender cover traffic with real messages is not an effective strategy to counter SDA.

V. CONCLUSIONS AND FUTURE WORK

Statistical disclosure [4], [9] is known to be a powerful long-term attack against a mix [3] that intends to make

possible anonymous communication between senders and receivers connected to it. We first presented a way to strengthen this attack by employing a weighted mean of the attacker's observations and showed by an example that this makes the attack more accurate than that of [9], which employs arithmetic mean. We then showed that dummy messages generated by senders as cover traffic for their real messages is not an effective strategy to counter such an attack. Despite the presence of such cover, the attacker has enough information to expose, over time, the receivers a particular sender mostly sends its messages to. Our mathematical analysis substantiates the empirical findings of [8], which lacked the rationale behind the ineffectiveness of this seemingly adequate strategy.

In the model of the anonymity system considered in this paper, all dummy cover traffic is generated by the senders, and is identified and blocked by the mix. Also, all real messages entering the mix are transmitted to their receivers in the same round. Several variations of this model are possible with a view to increasing the effectiveness of the countermeasure against the attack. First, it is possible for the mix to not block the cover traffic but send it to the receivers. Second, as studied in [8], cover traffic may be generated by the mix instead of the senders. Third, a cover traffic strategy can be combined with some message delaying methods of, for instance, [5] and [6]. In any countermeasure where dummy messages reach receivers, there is scope for intelligently targeting such messages to make determination of the \vec{O} and \vec{u} vectors, thus \vec{v} as well, more difficult.

Some of these variations have been studied, again mostly by simulation experiments. In future, we plan to demonstrate their properties mathematically.

REFERENCES

- [1] A. Back, U. Möller and A. Stiglic, Traffic analysis attacks and tradeoffs in anonymity providing systems. In *Proceedings of the 4th International Workshop on Information Hiding*, pages 245–257, Pittsburgh, USA, April 2001.
- [2] O. Berthold and H. Langos, Dummy traffic against long-term intersection attacks. In *Proceedings of the 2nd Privacy Enhancing Technologies Workshop*, San Francisco, USA, April 2002.
- [3] D. Chaum, Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, vol. 24, no. 2, pages 84–88, 1981.
- [4] G. Danezis, Statistical Disclosure Attacks: Traffic confirmation in open environments. In *Proceedings of Security and Privacy in the Age of Uncertainty*, pages 421–426, Athens, May 2003.
- [5] C. Díaz and A. Serjantov, Generalising Mixes. In *Proceedings of the 3rd Privacy Enhancing Technologies Workshop*, pages 18–31, Dresden, Germany, March 2003.

- [6] D. Kesdogan, J. Egner and R. Büschkes, Stop-and-go MIXes: Providing probabilistic anonymity in an open system. In *Proceedings of the International Workshop on Information Hiding*, April 1998.
- [7] D. Kesdogan, D. Agrawal and S. Penz, Limits of anonymity in open environments. In *Proceedings of the 5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, October 2002.
- [8] N. Malleš and M. Wright, Countering statistical disclosure with receiver-bound cover traffic. In *Proceedings of the 12th European Symposium on Research In Computer Security*, pages 547–562, Dresden, Germany, 2007.
- [9] N. Mathewson and R. Dingledine, Practical traffic analysis: Extending and resisting statistical disclosure. In *Proceedings of the 4th Privacy Enhancing Technologies Workshop*, pages 17–34, Toronto, Canada, May 2004.
- [10] J.-F. Raymond, Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In *Designing Privacy Enhancing Technologies: Proceedings of International Workshop on Design Issues in Anonymity and Unobservability*, pages 10–29, 2001.
- [11] A. Serjantov, R. Dingledine and P. Syverson, From a trickle to a flood: Active attacks on several mix types. In *Proceedings of the 5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, October 2002.
- [12] V. Shmatikov and M.-H. Wang, Timing analysis in low-latency mix networks: attacks and defenses. In *Proceedings of the 11th European Symposium on Research in Computer Security*, pages 18–33, Hamburg, Germany, September 2006.