SENDER COVER TRAFFIC TO COUNTER AN IMPROVED STATISTICAL DISCLOSURE ATTACK

A Thesis by

Huabo Lu

B.E., Beijing Forestry University, China, 2008

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

July 2011

SENDER COVER TRAFFIC TO COUNTER AN IMPROVED STATISTICAL DISCLOSURE ATTACK

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Computer Networking.

Rajiv Bagai, Committee Chair

Bin Tang, Committee Member

Tianshi Lu, Committee Member

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Rajiv Bagai, who made it possible for me to complete this thesis. His support, knowledge, and patience have guided me from the very beginning to the end. I would also like to thank Dr. Bin Tang and Dr. Tianshi Lu, for their kind help and for serving on my thesis committee.

ABSTRACT

The statistical disclosure attack (SDA) is quite an effective method for attackers to go against an anonymity system and to reveal the information behind it. It targets at a particular anonymity system user, tries to find its sending/receiving relationship with others after a long term observation. In this thesis, we first make an improvement for SDA, using weighted mean instead of the orginal arithmetic mean to calculate the cumulative observed receiver popularity in order to get a more precise value of the attack result. Second, we present an analysis for the effectiveness of the sender generated cover traffic, showing that employing this kind of cover traffic helps little on protecting anonymity system users against a sophisticated attacker. The analysis also validates an earlier empirical result of the ineffectiveness of sender generated cover traffic.

TABLE OF CONTENTS

Chapter **Page**

1 INTRODUCTION 1

1.1 Thesis Contribution 2

1.2 Thesis Organization 2

2 LITERATURE REVIEW 4

2.1 Statistical Disclosure Attack 4

2.2 Cover Traffic Strategies 7

3 AN IMPROVED SDA 9

3.1 Proof of the Effectiveness of the Improved SDA Using Weighted Mean . . . 11

4 SENDER COVER TRAFFIC 14

5 CONCLUSIONS AND FUTURE WORK 19

REFERENCES 20

LIST OF FIGURES

Figure **Page**

2.1 Message flow in round k 5

4.1 Message flow with dummy messages in round k 16

CHAPTER 1

INTRODUCTION

The current Internet's framework provides great convenience for people to communicate and exchange data with each other. However, the growing needs for anonymous communication, such as anonymous web surfing, report emailing, evoting and some various applications that require anonymity, cannot be satisfied directly by Internet's framework. In order to satisfy the need for anonymity, people have to design various protocols on top of the current Internet architecture and then send application data through them.

The requirement for anonymity system could be satisfied by a kind of implementation, i.e., the *mix* network, first proposed by Chaum in [1]. The mix network is a quite popular technique and has been studied widely. It is a collection of one or more proxy nodes, which carry messages (network traffic) between senders and, possibly overlapping, receivers connected to the mix. These intermediate proxies are essential building blocks for this kind of anonymity system.

The study around mix-based anonymity system also includes several possible attacks and their countermeasures, if exists. Back, Möller and Stiglic in [2], and Raymond in [3] contain detailed lists and descriptions of attacks. Of all these attacks, the class of long-term *intersection attacks* is one of the most powerful. In these attacks, a quiet, yet powerful observer (also called a passive global attacker in some text) would try to link *senders* with their corresponding *receivers* based on their long-term observation of the messages that enter and exit the mix network.

One member of long-term intersection attacks, the *Statistical Disclosure Attack (SDA)*, is in the range of our study. It is proposed by Danezis in [4] and aims at a single particular sender. The attacker that performs SDA keeps observing the relative popularity of various receivers in the anonymity system, and aims to reveal some receivers that are related to that particular sender from all receivers. Mathewson and Dingledine present an extended version

of SDA in [5] by reducing the original model's restrictions and list some possible defense methods for this attack like cover (dummy) traffic.

1.1 Thesis Contribution

Our first contribution in this thesis is an improved version of the extended SDA of [5]. We claim that the *weighted* mean of the observed relative popularity of the receivers is more accurate than one obtained by *arithmetic* mean. Besides an example in Chapter 3, we also provide the proof in Section 3.1.

Sending cover traffic (also called dummy traffic sometime) along with real ones is a strategy to counter SDA that has been studied in the past. The cover traffic aims at confusing the possible attacker so that it could not make a precise observation and furthermore could be thwarted. It is designed that senders will generate the cover traffic and the mix node will detect and block it. However, Mallesh and Wright in [6] shows empirical results indicating the cover traffic generated by senders could be ineffective. They set experiments simulating the scenario of extended SDA with cover traffic, resulting that the attacker succeeds despite the existence of cover traffic.

The second contribution of this thesis is a mathematical analysis of the impact of sender generated cover traffic to the SDA, in the model of mix-based anonymity system. The analysis supports the experimental findings in [6], i.e., SDA is not significantly affected by sender cover traffic. The results contained in this thesis appeared in Bagai et al. [7] and Tang et al. [8].

1.2 Thesis Organization

The rest of thesis is formed as follows. In Chapter 2 we take a quick overview of the extended SDA in [5] which is based on the basic anonymous network model containing real messages only. It shows details about how a quiet powerful observer can determine the set of receivers of a particular sender after a long-term observation. In Chapter 3 we present our improvement to the attack described, i.e., using weighted mean to calculate the receiver popularity. We show, by an example and a proof, that this results in an attack that is more

accurate than that of [5], which uses only arithmetic mean. In Chapter 4 we first extend the basic model to a model that allows senders generating cover traffic and blending them into real ones. Then we show how the attack can perform almost unhindered despite the existence of such cover traffic, which is in accordance with the result in [6]. In Chapter 5 we concludes our main results and present some possible directions for future work.

CHAPTER 2

LITERATURE REVIEW

In this chapter we show a review of SDA, which is proposed by Danezis in [4]. Later, Mathewson and Dingledine show an extended version in [5]. The original SDA and the extended one are based upon the disclosure attack, which is proposed by Kesdogan et al. in [9], have been well studies, as in Mallesh and Wright in [6]. We also give an overview of three cover traffic strategies, mentioned in [6], for countering this attack.

2.1 Statistical Disclosure Attack

The model for SDA is based on a mix network, as introduced by Chaum in [1]. The mix network could contain one or more mix node, while an attacker could treat them as one node once it can observe the incoming and outgoing traffic on the network edge. Senders and receivers are connected to mix network, whose job is to transmit messages sent by any senders to their destined receivers without exposing its orignal sender. Anonymity is vulnerable to timing analysis by a global observer if the mix network transmit messages immediately. Therefore, the mix collects a certain number (fixed or dynamic) of messages in each *round*, and send them out simultaneously. Adding some other techniques like re-coding, the mix will provide protection against bit pattern comparison attack. Such round repeats when the mix is intended to function.

The target of SDA is a particular sender, called *Alice*. The aim of the attack is to reveal Alice's friends, i.e., the subset of receivers that Alice sends messages to, over a period of time. This aim is achieved by observing the messages entering and exiting the mix at every round. All other senders, beside Alice, are called *background* senders.

We use a_k to denote the number of messages sent by Alice in round k, and b_k to denote the number of messages sent by all other senders in that round. Thus, in that round there will be $a_k + b_k$ messages entering and exiting the mix. Figure 2.1 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.

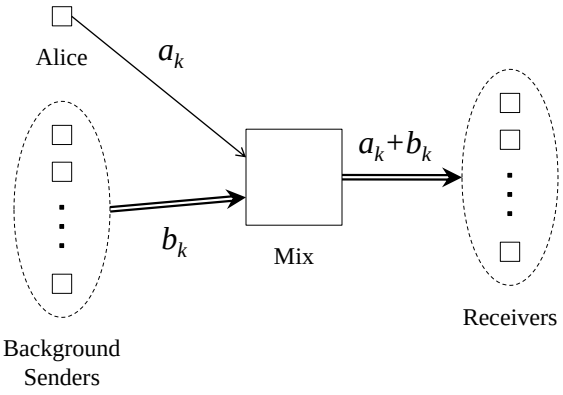

Figure 2.1: Message flow in round k

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

$$\vec{o_k}[i] = \frac{\vec{r_k}[i]}{a_k + b_k}$$

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

$$\vec{v} \approx \frac{\bar{n}\vec{O} - (\bar{n} - \bar{m})\vec{u}}{\bar{m}}. \tag{2.1}$$

All values on the right side of equation (2.1) can be obtained by observing the mix over time, thus making possible a reasonable estimate of the receivers' degrees of friendship with Alice.

2.2 Cover Traffic Strategies

As shown in the previous section, SDA is quite powerful in revealing a specified user's sending profile after the observation of many rounds. In contrast to normal user traffic, cover traffic, or dummy traffic, contains trivial information and will be dropped somewhere (for example, at the mix node or receiver). The aim of cover traffic is to confuse the possible attacker so that one or more parameters in the attacker's SDA calculation will be impacted, further slowing down or thwarting SDA by affecting the correctness of the user's sending profile. Mallesh and Wright in [6] and Mathewson and Dingledine in [5] present some cover traffic strategies in detail.

Based on the SDA model, there are three types of cover traffic:

User cover traffic: User cover traffic is generated by the user and is dropped at the mix node. Usually, communication between users and the mix node is encrypted, so it is easy for the mix node to distinguish the real traffic from cover traffic after decryption while possible attackers can hardly do so. The purpose of user cover traffic is to block the attacker from knowing the true number of traffics sent by "Alice" in a round, and thus impacting the calculation of $\bar{m}$.

Background cover traffic: Once the attacker locks on "Alice" as its target, the user cover traffic sent by other senders will be considered as background cover traffic. As with user cover traffic, it will be dropped by the mix node. This cover traffic makes it more

difficult for the attacker to obtain the correct average number of messages sent in a round, i.e., $\bar{n}$.

Mix-generated cover traffic: In addition to the general function, the mix node can generate cover traffic as well. Mix-generated cover traffic travels from mix node to final receiver and is dropped there. It is common for this type of cover traffic to not be encrypted, due to the service that the final receiver provides (for example, it is common that a search query is not encrypted). This character requires more work to make it indistinguishable from normal traffic. [6] provides some detailed proposals to work that out. Mix-generated cover traffic can hinder the precision of $\vec{o_i}$, and $\vec{O}$, and further impact the calculation of $\vec{o_i}$.

It is possible to use one type of cover traffic alone or use several types of cover traffic in combination at the same time. Empirical results [6] show that among all three cover traffic strategies, only the mix-generated cover traffic is effective to thwarting SDA. A mathematic analysis for the ineffectiveness of user-generated cover traffic is provided in Chapter 4.

CHAPTER 3

AN IMPROVED SDA

In the original SDA proposed by Danezis [4], the mix outputs a constant number of messages in each round, exactly one of which is sent from Alice. For this model, calculating $\vec{O}$ as an arithmetic mean of the $\vec{o_k}$ vectors is all that is needed. Another reason for using arithmetic mean in that model is that $\vec{u}$ corresponds to uniform distribution over all receivers, which is fixed and does not need to be computed once observed by attacker.

Mathewson and Dingledine in [5] present a model that is an extension of the original SDA. In that model, the number of messages transmitted by the mix in each round can vary, Alice is allowed to send multiple messages in any round, and $\vec{u}$ need not be uniform. However, the SDA of [5] continues to use the same arithmetic mean method for calculating $\vec{O}$ (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.

Here we give 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 calculation 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

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

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 send any messages.

In order to better study the effect of this change of calculation 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 3.1.

Table 3.1: Messages sent to receivers A and B

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

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 3.1 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 (2.1), 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 $\langle 0.83, 0.17 \rangle$.

3.1 Proof of the Effectiveness of the Improved SDA Using Weighted Mean

We begin with the following definitions:

- $\vec{r_k^A}$: the vector containing the number of Alice's messages received by each receiver in round k.
- $\vec{r_k^B}$: the vector containing the number of background senders' messages received by each receiver in round k.
- $\vec{v}_{real}[i]$: the receiver i's actual relative degree of friendship to Alice after t rounds:

$$\vec{v}_{real}[i] = \frac{\sum_{k=1}^{t} r_k^A[i]}{\sum_{k=1}^{t} a_k}.$$

- $\vec{O}_w[i]$ and $\vec{O}_a[i]$: the cumulative observed relative popularity for receiver i by weighted mean and arithmetic mean, respectively. From Section 2.1 and Chapter 3, we have:

$$\vec{O}_w[i] = \frac{\sum_{k=1}^{t} \vec{r_k}[i]}{\sum_{k=1}^{t}(a_k + b_k)}, \tag{3.1}$$

$$\vec{O}_a[i] = \frac{\sum_{k=1}^{t} \vec{o_k}[i]}{t} = \frac{\sum_{k=1}^{t} \frac{\vec{r_k}[i]}{a_k+b_k}}{t}. \tag{3.2}$$

- $\vec{u}_w[i]$ and $\vec{u}_a[i]$: the cumulative observed relative popularity with the background senders, in rounds when Alice does not send messages, by weighted mean and arithmetic mean respectively.

- $\vec{v}_w[i]$ and $\vec{v}_a[i]$: the receiver i's relative degree of friendship to Alice after t rounds, obtained by weighted mean and arithmetic mean respectively:

$$\vec{v}_w[i] \approx \frac{\overline{n}\vec{O}_w[i] - (\overline{n} - \overline{m})\vec{u}_w[i]}{\overline{m}}, \tag{3.3}$$

$$\vec{v}_a[i] \approx \frac{\overline{n}\vec{O}_a[i] - (\overline{n} - \overline{m})\vec{u}_a[i]}{\overline{m}}. \tag{3.4}$$

By substituting (3.1) into (3.3) and (3.2) into (3.4), we get

$$\vec{v}_w[i] \approx \frac{\sum_{k=1}^{t} \vec{r_k}[i] - \sum_{k=1}^{t} b_k \times \vec{u}_w[i]}{\sum_{k=1}^{t} a_k}, \tag{3.5}$$

$$\vec{v}_a[i] \approx \frac{\sum_{k=1}^{t}(a_k + b_k) \sum_{k=1}^{t} \frac{\vec{r_k}[i]}{a_k+b_k} - \sum_{k=1}^{t} b_k \times \vec{u}_a[i]}{\sum_{k=1}^{t} a_k}. \tag{3.6}$$

To show that by employing a weighted mean of the observed relative receiver popularity, the attacker can determine more accurately the set of receivers that a user sends messages to than using existing arithmetic mean-based one, we show that this is the case for each receiver i.

Theorem. For any receiver i, $|\vec{v}_{real}[i] - \vec{v}_w[i]| \leq |\vec{v}_{real}[i] - \vec{v}_a[i]|$.

Proof. We need to show that $(\vec{v}_{real}[i] - \vec{v}_w[i])^2 \leq (\vec{v}_{real}[i] - \vec{v}_a[i])^2$, i.e.

$$2 \times \vec{v}_{real}[i] \times (\vec{v}_a[i] - \vec{v}_w[i]) \leq (\vec{v}_a[i] - \vec{v}_w[i]) \times (\vec{v}_a[i] + \vec{v}_w[i]). \tag{3.7}$$

We first show that $\vec{v}_a[i] - \vec{v}_w[i] > 0$, and then we will only need to prove $2 \times \vec{v}_{real}[i] \leq \vec{v}_a[i] + \vec{v}_w[i]$.

From (3.6) and (3.5),

$$\begin{aligned}
&\vec{v}_a[i] - \vec{v}_w[i] \\
&= \frac{(\sum_{k=1}^{t}(a_k+b_k)\sum_{k=1}^{t}\frac{\vec{r_k}[i]}{a_k+b_k} - \sum_{k=1}^{t} b_k \times \vec{u}_a[i]) - (\sum_{k=1}^{t}\vec{r_k}[i] - \sum_{k=1}^{t} b_k \times \vec{u}_w[i])}{\sum_{k=1}^{t} a_k} \\
&= \frac{(\sum_{k=1}^{t}(a_k+b_k)\sum_{k=1}^{t}\frac{\vec{r_k}[i]}{a_k+b_k} - \sum_{k=1}^{t}\vec{r_k}[i]) + \sum_{k=1}^{t} b_k \times (\vec{u}_w[i] - \vec{u}_a[i])}{\sum_{k=1}^{t} a_k}
\end{aligned}$$

We have that

$$\begin{aligned}
&\sum_{k=1}^{t}(a_k+b_k)\sum_{k=1}^{t}\frac{\vec{r_k}[i]}{a_k+b_k} \\
&= \sum_{k=1}^{t}(a_k+b_k) \times \frac{\vec{r_1}[i]}{a_1+b_1} + \sum_{k=1}^{t}(a_k+b_k) \times \frac{\vec{r_2}[i]}{a_2+b_2} + ... + \sum_{k=1}^{t}(a_k+b_k) \times \frac{\vec{r_k}[i]}{a_k+b_k} \\
&> \vec{r_1}[i] + \vec{r_2}[i] + ... + \vec{r_t}[i] \\
&= \sum_{k=1}^{t}\vec{r_k}[i].
\end{aligned}$$

Furthermore, since the observed receiver popularity with background senders does not depend on whether the attacker uses the weighted mean or arithmetic mean, we can assume that $\vec{u}_w[i] = \vec{u}_a[i]$. Therefore, we have that $\vec{v}_a[i] - \vec{v}_w[i] > 0$, and from (3.7), we only need to prove $\vec{v}_a[i] + \vec{v}_w[i] \geq 2 \times \vec{v}_{real}[i]$, shown as below:

$$\begin{aligned}
&\vec{v}_a[i] + \vec{v}_w[i] \\
&= \frac{(\sum_{k=1}^{t}(a_k+b_k)\sum_{k=1}^{t}\frac{\vec{r_k}[i]}{a_k+b_k} - \sum_{k=1}^{t} b_k \times \vec{u}_a[i]) + (\sum_{k=1}^{t}\vec{r_k}[i] - \sum_{k=1}^{t} b_k \times \vec{u}_w[i])}{\sum_{k=1}^{t} a_k} \\
&> \frac{(\sum_{k=1}^{t}\vec{r_k}[i] + \sum_{k=1}^{t}\vec{r_k}[i]) - \sum_{k=1}^{t} b_k \times (\vec{u}_w[i] + \vec{u}_a[i])}{\sum_{k=1}^{t} a_k} \\
&= 2 \times \frac{\sum_{k=1}^{t}\vec{r_k}[i] - \sum_{k=1}^{t} b_k \times \vec{u}_w[i]}{\sum_{k=1}^{t} a_k} \\
&= 2 \times \frac{\sum_{k=1}^{t}\vec{r_k}[i] - \sum_{k=1}^{t} r_k^B[i]}{\sum_{k=1}^{t} a_k} \\
&= 2 \times \frac{\sum_{k=1}^{t} r_k^A[i]}{\sum_{k=1}^{t} a_k} \\
&= 2 \times \vec{v}_{real}[i] \qquad \square
\end{aligned}$$

CHAPTER 4

SENDER COVER TRAFFIC

The well known SDA is a very powerful attack that, once given time, quite accurately accomplishes its goal, i.e., revealing a particular sender's friends. Meanwhile, lots of 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 cover traffic that appear to the attacker just like real traffic.

There are several strategies for introducing additional delay for messages. Kesdogan, Egner and Büschkes proposed stop-and-go mixes in [10] that the mix node will hold any incoming message within it according to the acceptable message latency specified by sender. Exit times for messages also get extended in batching strategies by Serjantov, Dingledine and Syverson in [11]. Pool mixes of Díaz and Serjantov in [12] incorporate a distribution function to determine how to tailor the anonymity/delay tradeoff that adapts to traffic load fluctuations.

However, the effectiveness of the message delaying countermeasure is not strong. Mathewson and Dingledine in [5] study the effect of these pool mixes specifically for thwarting SDA. As all values in equation (2.1) 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.

On the other hand, the strategy of adding cover traffic blending into the real traffic is somehow more effective. Berthold and Langos propose a method in [13] for sending cover traffic on Alice's behalf when she is inactive, but Mathewson and Dingledine in [5] mention many problems associated with that approach. Shmatikov and Wang in [14] propose another method that asking senders to send cover traffic to the mix node in advance, and the mix

node will use them when needed later. Although this method is for low-latency networks, it does not work well for long-term intersection attacks, such as SDA. Mallesh and Wright [6] study the effects of both sender-generated and mix-generated cover traffic on SDA. They point out via simulation that sender-generated cover traffic is not effective against SDA, while mix-generated is quite effective. We now substantiate their simulation results for the sender-generated cover traffic via a mathematical argument for the ease with which SDA can be carried out almost unhindered.

We consider the model that allow all senders (including Alice) to send cover traffic to the mix. As an assumption of SDA is that external observers cannot see the content of message. In this case, these cover traffic are indistinguishable from the real ones to such an observer. However, the mix has the ability to tell the cover traffic 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 (2.1).

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 (2.1), 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 4.1 shows the message flow that includes dummy messages via the mix in round k.

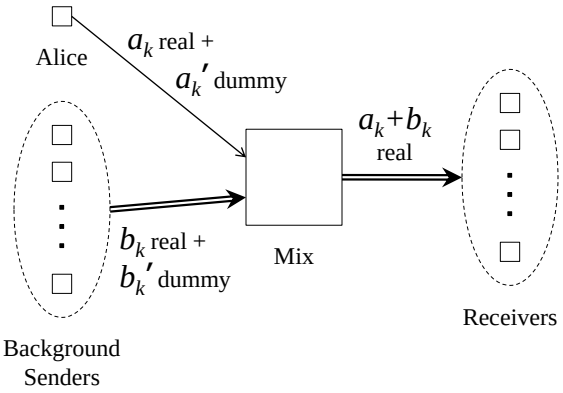

Figure 4.1: Message flow with dummy messages in round k

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 (2.1) 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.

$$\begin{aligned} \alpha &= \frac{a'_{k_1}}{a_{k_1}} = \frac{a'_{k_2}}{a_{k_2}}, \text{ and} \\ \beta &= \frac{b'_{k_1}}{b_{k_1}} = \frac{b'_{k_2}}{b_{k_2}}, \text{ for all } k_1 \text{ and } k_2. \end{aligned}$$

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 (2.1), 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.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

Statistical disclosure attack [4, 5] is known to be a powerful long-term attack against a mix [1] network that intends to provide anonymity Internet communication between senders and receivers connected to it. We first proposed a way to enhance SDA by using a weighted mean of the attacker's observations and showed by an example that this makes the attack more accurate than that of [5], which uses arithmetic mean. We then showed that cover traffic generated by senders as a countermeasure to SDA is not an effective strategy. Despite the presence of such cover, the attacker still can get sufficient information to expose, over time, the receivers a particular sender mostly sends its messages to. Our mathematical analysis substantiates the empirical findings of [6], which lacked the rationale behind the ineffectiveness of this seemingly adequate strategy.

In the model of the anonymity system considered in this thesis, 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 [6], 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, [12] and [10]. 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

LIST OF REFERENCES

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

[2] 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.

[3] 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.

[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] 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.

[6] N. Mallesh 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.

[7] R. Bagai, H. Lu and B. Tang, On the sender cover traffic countermeasure against an improved statistical disclosure attack. In *Proceedings of the 8th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC'10)*, pages 555–560, Hong Kong, China, 2010.

[8] B. Tang, R. Bagai and H. Lu, An improved statistical disclosure attack. Submitted to *Information Processing Letters*. 2011.

[9] 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.

[10] 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.

[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.

LIST OF REFERENCES (continued)

[12] C. Díaz and A. Serjantov, Generalising Mixes. In *Proceedings of the 3rd Privacy Enhancing Technologies Workshop*, pages 18–31, Dresden, Germany, March 2003.

[13] 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.

[14] 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.