Reasoning about the Anonymity Provided by Pool Mixes that Generate Dummy Traffic

Claudia Díaz and Bart Preneel

K.U.Leuven Dept. Electrical Engineering-ESAT/COSIC
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
claudia.diaz@esat.kuleuven.ac.be, bart.preneel@esat.kuleuven.ac.be
http://www.esat.kuleuven.ac.be/cosic/

Abstract. In this paper we study the anonymity provided by generalized mixes that insert dummy traffic. Mixes are an essential component to offer anonymous email services. We indicate how to compute the recipient and sender anonymity and we point out some problems that may arise from the intuitive extension of the metric to take into account dummies. Two possible ways of inserting dummy traffic are discussed and compared. An active attack scenario is considered, and the anonymity provided by mixes under the attack is analyzed.

1 Introduction

The Internet was initially perceived as a rather anonymous environment. Nowadays, we know that it is a powerful surveillance tool: anyone willing to listen to the communication links can spy on you, and search engines and data mining techniques are becoming increasingly powerful. Privacy does not only mean confidentiality of the information; it also means not revealing information about who is communicating with whom. Anonymous remailers (also called *mixes*) allow us to send emails without disclosing the identity of the recipient to a third party. They also allow the sender of a message to stay anonymous towards the recipient.

In this paper, we extend previous results [DS03b,SD02,DSCP02] in order to obtain equations to compute sender and recipient anonymity, expressed using the model of generalised mixes. Then, we reason about the anonymity provided by these mixes when dummy traffic is inserted in the network. We point out that the intuitive way of computing the anonymity when dummy traffic is inserted by the mix presents some problems. We also analyze the anonymity offered by the mixes when an active attacker is capable of deploying an $n-1$ attack. Some side aspects are discussed, in order to provide a good understanding of the anonymity metric. The paper also intends to be an intermediate step towards the quantification of the anonymity provided by the whole mix network.

The structure of the paper is as follows: in Sect. 2 we give an overview on mixes. In Sect. 3 the concept of dummy traffic is introduced. Anonymity metrics are discussed in Sect. 4. Sections 5 and 8 provide results for recipient anonymity,

first without dummy traffic and then with dummy traffic. Sender anonymity is analyzed in Sect. 6 and Sect. 7. Sect. 9 analyzes recipient anonymity under an active attack. Finally, Sect. 10 presents the conclusions and proposes topics of future work.

2 Mixes

Mixes are the essential building block to provide anonymous email services. A mix is a router that hides the correspondence between incoming and outgoing messages. A taxonomy of mixes can be found in [DP04]. The mix changes the appearance and the flow of the messages. In order to change the appearance of the messages, the mix uses some techniques, such as padding and encryption, thus providing bitwise unlinkability between inputs and outputs. Techniques like reordering and delaying messages, and generating dummy traffic are used to modify the flow of messages. This modification of the traffic flow is needed to prevent timing attacks that could disclose the relationship between an input and an output messages by looking at the time the message arrived to and left from the mix.

The idea of mixes was introduced by Chaum [Cha81]. This first design was a *threshold mix*, a mix that collects a certain number of messages and then flushes them. Since then, variants on this first design have been proposed in the literature [DS03b,MC00,Cot,Jer00]. One of the design strategies used to increase the anonymity of the messages and prevent some simple attacks is sending only part of the messages, while keeping others for later rounds. These are called *pool mixes* or *batching mixes*. Chaum's original design is a particular case of a pool mix, that keeps 0 messages in the pool when it flushes.

Another type of mixes, *synchronous* or *Stop-and-Go* mixes, were proposed by Kesdogan *et al.* in [KEB98]. These mixes modify the traffic flow just by delaying messages. They cannot be expressed as generalized mixes [DS03b], and their analysis is outside the scope of this paper. Some practical measurements on continuous mixes have been presented by Díaz *et al.* in [DSD04].

2.1 Generalized mixes

The concept of generalized mixes was introduced by Díaz and Serjantov in [DS03b]. Here, we summarize the basic concepts of the generalized mixes model. Pool mixes are expressed in this model by a function, instead of a detailed algorithm. The mix is represented at the time of flushing, making abstraction of the event that triggers the flushing: it may be the expiration of a timeout (*timed mixes*) or the arrival of a message (*threshold mixes*). However, in Sect. 4.1 we point out some properties of threshold mixes which are worth discussing.

A *round* represents a cycle of the mix; during a round, the mix collects input messages that are placed in the pool, the last event of the round is the flushing of messages. The function $P(n)$ represents the probability of the messages being sent in the current round, given that the mix contains n messages in the pool.

An example of a timed pool mix that keeps 20 messages in the pool and flushes the rest is shown in Fig. 1. In this case: $P(n) = 0$ for $n \leq 20$ and $P(n) = 1 - 20/n$ for $n > 20$.

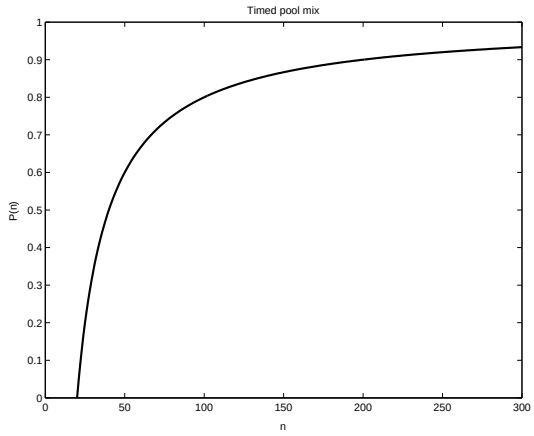

Fig. 1. Representation of a generalized mix

Note that all messages contained in the pool have the same chance of being selected for sending. This uniformity maximizes the randomness of the selection. Reducing this randomness leads to leaking more information about the outputs.

2.2 Deterministic vs. binomial mixes

$P(n)$ denotes the probability of sending every message. There are two ways of dealing with this probability. We distinguish between *deterministic* and *binomial* mixes. Note that the value of the function $P(n)$ is independent of the mix being deterministic or binomial.

Deterministic mixes. If a mix is deterministic then the number of messages sent is determined by the number of messages contained in the pool; the mix sends $s = nP(n)$ messages. The only randomness present in the flushing algorithm is the one used to select *which* messages will be sent, but not *how many*. Classical pool mixes fall into this category. Note that, for these mixes, once the number of messages in the pool (n) is known, the number of messages sent (s) is determined, and vice versa.

Binomial mixes. Binomial mixes were introduced in [DS03b]. In these mixes, an independent decision is taken for every message in the pool. A biased coin (being the bias the value of $P(n)$) is thrown for each message, so it is sent with

probability $P(n)$. The number of selected messages follows a binomial distribution with respect to the number of messages in the pool. The probability of sending s messages, given that the pool contains n messages is (note that p is the result of the $P(n)$ function for the current round):

$$Pr(s|n) = \frac{n!}{s!(n-s)!} \cdot p^s \cdot (1-p)^{n-s} \quad .$$

The probability of having n messages in a pool of maximum size N_{max}, given that the mix sends s messages is [DS03b]:

$$Pr(n|s) = \frac{p(s|n)}{\sum_{i=s}^{N_{\max}} p(i|n)} \quad .$$

This probabilistic relationship has the following effects: as it was shown in [DS03b], just by observing the number of outputs of a round, an observer cannot know *exactly* the number of messages contained in the mix; by knowing the number of messages in the pool we cannot determine the number of messages that will be flushed. However, large deviations from the mean values occur with very low probability. This property influences the anonymity metric under certain circumstances, as it is remarked in Sect. 4.2.

3 Dummy traffic

Inserting dummy traffic (see [Jer00] for a discussion on the topic and [DP04] for a taxonomy of dummy traffic) in a mix network is a technique that hides the traffic patterns inside the mix network, making traffic analysis more difficult. As shown in Sect. 8, the generation of dummy traffic increases the anonymity of the messages sent through the mix network. Dummies also reduce the latency of the network by keeping a minimum traffic load (a low traffic load increases the latency of the mix network).

A dummy message is a "fake" message created by a mix, instead of a user. The final destination is also a mix, instead of a recipient; the dummy is discarded by the last mix, that may be the one that created it. Observers of the network and intermediate mixes cannot distinguish the dummy from a real message. In this paper, we make abstraction of the specific purpose of the dummy (link padding, ping traffic, etc.) and its path-length; we focus on the impact of these dummies in the anonymity provided by the mix that creates the dummies (note that dummies are treated as real messages by the other mixes, except for the last in the path, that discards them).

Creating and transmitting dummies has a cost. We need to find a tradeoff between the anonymity we want to offer and the cost of adding dummy traffic. In this paper we present formulas to compute the anonymity, taking into account the number of dummies produced by mixes. One possibility is that the dummies created by a mix are sent to itself through a path in the network. Therefore, every mix will discard its own dummies, and no mix is able to distinguish real messages from the dummies created by another mix. This strategy was already

proposed by Danezis and Sassaman in [DS03a] in order to detect and prevent active attacks against a mix.

We assume that the mix generates dummies following a probability distribution. The creation of a fixed number d of dummies per round is a particular case, in which $\Pr(d_k = d) = 1$. The probability distribution that determines the number of dummies created should be independent of the traffic of real messages. Otherwise, an active attacker could develop an attack strategy which minimizes the number of dummies sent during his attack.

We consider two possible scenarios. First, we assume that the mix inserts the dummy messages into the output link at the time of flushing. If the mix flushes after a timeout (timed mix), the mix could add dummies even in the case in which no real messages are sent. In this case, the pool contains only real messages (note that dummies created by other mixes are considered real messages at the intermediate mixes).

In the second scenario, a number of dummies is added to the pool of the mix. In this case, the number of dummies present at the output depends on the random selection of messages from the pool. The function $P(n)$, that defines the probability with which messages are going to be sent, is computed taking into account the dummies present in the pool. Otherwise, in the case of low traffic, the mix would accumulate dummies that are flushed at a very low rate. Besides, the goal of keeping traffic above a minimum would not be achieved.

We also assume that the number of inserted dummies is independent of the number of dummies already present in the pool, in order to keep the mix design *stateless*, that is, that the decisions of one round are not constrained by the events of previous rounds. A setting in which the mix keeps, for instance, a constant number of dummies in the pool would need a different analysis.

4 Anonymity metrics

In this section we introduce the anonymity metrics for mixes. We remark the particularities of some mix designs (binomial mixes and threshold mixes). Also, we present the attack model considered.

Anonymity was defined by Pfitzmann and Köhntopp [PK00] as *"the state of being not identifiable within a set of subjects, the anonymity set"*.

The use of the information theoretical concept of entropy as a metric for anonymity was simultaneously proposed by Serjantov and Danezis in [SD02] and by Díaz *et al.* in [DSCP02]. The difference between the two models for measuring anonymity is that in [DSCP02] the entropy is normalized with respect to the number of users. In this paper we will use the non-normalized flavour of the metric.

The anonymity provided by a mix can be computed for the incoming or for the outgoing messages. We call this *sender anonymity* and *recipient anonymity*.

Sender anonymity. In order to compute the sender anonymity, we want to know the effective size of the anonymity set of senders for a message output by the mix.

Therefore, we compute the entropy of the probability distribution that relates an outgoing message of the mix (the one for which we want to know the anonymity set size) with all the possible inputs.

Recipient anonymity. If we want to compute the effective recipient anonymity set size of an incoming message that goes through the mix, we have to compute the entropy of the probability distribution that relates the chosen input with all possible outputs.

Note that in the two cases, the metric computes the anonymity of a *particular* input or output message; it does not give a general value for a mix design and it is dependent on the traffic pattern. The advantage of this property is that mixes may offer information about the *current* anonymity they are providing. The disadvantage is that it becomes very difficult to compare theoretically different mix designs. Nevertheless, it is possible to measure on real systems (or simulate) the anonymity obtained for a large number of messages and provide comparative statistics. This has been done by Díaz *et al.* in [DSD04], where we can see that the anonymity offered by a mix can be analyzed through simulations.

4.1 Remarks on threshold mixes.

If an active attack is deployed (see Sect. 9), the attacker is able to empty the mix of previous messages much faster, because he is able to trigger the flushings by sending many messages. Also, the attacker may have another advantage: when a dummy arrives to the last mix of the path it is discarded and it does not trigger the flushing if only one message more is required to reach the threshold. This way, the attacker may be able to know whether a message is a dummy or not. For these reasons, timed mixes should be preferred to threshold mixes.

4.2 Remarks on binomial mixes

There are two ways of computing the anonymity metric for binomial mixes. If the number n_k of messages in the mix (at round k) and the number s_k of messages sent from the pool are observable, this information can be used in the computation of the entropy. We would use s_k/n_k instead of $P(n_k)$. The anonymity obtained is the one that corresponds to a particular realisation of the mix. Note that the same pattern of incoming traffic fed several times into a binomial mix may result in different values of the metric.

If this is not observable (dummy traffic can hide this number), or if we want to compute the *average*[1] anonymity offered by a mix, then we have to use the

[1] This *average* may be different of the one obtained by considering e possible scenarios (binomial output combinations), each of them providing an entropy $H_i, (i = 0 \ldots e)$, happening with probability $p_i, (i = 0 \ldots e)$. We have checked on a simple numerical example that the average entropy that we obtain by summing the entropies H_i ponderated by their probabilities p_i is different from this *average*, that corresponds to the *a priori most probable case.*

a priori probability, $P(n)$. In this case, we obtain a fixed result for a given incoming traffic.

4.3 Attack model and dimensions of uncertainty

The anonymity metric computes the uncertainty about the sender or the recipient of a message, given that some information is available. We compute the metric from the point of view of an attacker, whose powers must be clearly specified.

The attacker considered in the paper is a *permanent global passive observer*. The attacker knows the number of messages that arrive to the mix in every round (a_k) and the number of messages sent by the mix in every round (s_k). We assume that the function of the mix $P(n)$ is publicly known. Moreover, the attacker "has always been there" and "will always be there", that is, the attacker knows the whole history of the mix. This way we give a lower bound for anonymity, given that an attacker with less power will only obtain less information, and the users will be more anonymous towards him. In Sect. 9 we consider an active attacker, capable of deploying an $n-1$ attack.

When the mix does not generate dummy traffic, the attacker has all the information needed to compute the anonymity (a_k, s_k and $P(n_k)$), because he can determine the number of messages in the pool, n_k. When the mix generates dummies, we can find some differences between deterministic and binomial mixes. If the mix is deterministic, then the attacker can find out n_k, regardless of the dummy policy. If the mix is binomial, then for a deterministic dummy policy he will also be able to determine n_k (note that the attacker is *permanent* and knows all the history). But for a random dummy policy the value n_k cannot be determined, and therefore $P(n_k)$ remains unknown. This means that the attacker cannot compute with certainty the anonymity of the messages. He may be able to estimate it; the estimation is more accurate when the number of dummies or the randomness of the dummy distribution decreases.

It is important to note that this uncertainty is, in most cases, independent of the anonymity provided by the mix. The cases in which this uncertainty increases the anonymity are indicated in the appropriate sections.

Another sort of uncertainty arises if the attacker starts observing the system when it has been running for some time (non permanent attacker), or if the mix starts with an unknown number of messages in the pool. This type of attacker has been considered in the literature (see, for example, [SN03]). In this case, the uncertainty about the number of unknown messages contained in the pool (arrived before the attacker started observing) decreases with every round, as the probability of any of them still being there does.

4.4 Anonymity provided by a mix network

In this paper, we compute the anonymity of a single mix. Nevertheless, we assume that the mix is a node of a mix network (otherwise, it would not make sense to create dummy traffic). The goal of the analysis of the impact of dummy traffic

on the anonymity provided by a mix is to go a step further towards a metric that computes the anonymity provided by a mix network, when dummy traffic is inserted by the nodes.

Without the results provided in this paper, it would not be clear the way of computing the anonymity of a mix network whose nodes insert dummy traffic. As we show in Sect. 7 and Sect. 8, we must be careful when applying the information theoretical anonymity metrics to mixes that generate or discard dummies.

Danezis [Dan03] has proposed a method to measure the anonymity provided by a mix network (in the absence of dummy traffic). The method can be applied to compute the recipient anonymity as follows: one measures the anonymity of a mix network as the entropy of the distribution of probabilities that relates a message m entering the network with all the possible outputs of the network, o_{ij} (being i the mix that outputs the message and j the message number). These probabilities are expressed as the product of two terms: first, the probability of the target input m being output o_{ij} conditioned to the fact that the m left at the same mix M_i as output o_{ij}; second, the probability of the target having been left from mix M_i.

The first term, $\Pr(m = o_{ij} | m \text{ left at } M_i)$ corresponds to the anonymity provided by mix M_i (i.e., the formulas presented in this paper are suited to compute this value). The second quantifies how effectively the traffic from different nodes is mixing together; it is dependent of the topology of the network and on the path selection of the messages and dummies. In order to effectively enhance the anonymity provided by the mix network, the dummy traffic should maximize the number and the probabilistic uniformity of the possible destinations for every outgoing message.

Although the computation of the second term when mixes create dummy traffic may not be obvious, the results provided in Sect. 8 and Sect. 7 may be useful to measure the impact of dummy traffic on anonymity at network scale.

5 Recipient anonymity without dummy traffic

In this section, we compute the effective recipient anonymity set size of an incoming message that goes through the mix. We need to compute the entropy of the probability distribution that relates the chosen input with all possible outputs.

We summarize the notation needed for this section:

- a_k: number of messages arrived to the mix in round k.
- n_k: number of messages in the mix in round k (before flushing).
- s_k: number of messages sent by the mix in round k.
- $P(n)$: characteristic function of a generalized mix [DS03b]. It represents the probability of a message that is in the pool of being flushed as a function of the number of messages contained in the mix.
- $p(O_i)$: probability of linking the chosen input with an output O that left the mix in round i.
- H_r: effective recipient anonymity set size. Also *recipient anonymity.*

Computing the recipient anonymity has a shortcoming: instead of needing the past history of the mix, we need to know the *future history*. In theory, we should wait infinite time before we can compute the entropy of an input. In practice, we can give an approximation of this value once the probability of the message still staying in the mix is very low (we can choose the probability to be arbitrarily small, and get as close to the real entropy as we want). Note that the approximation is still giving a lower bound for anonymity, because the approximated entropy is lower than the real one.

From [DS03b], we know that if a message arrived to the mix in round r, the probability of this message going out in round i is:

$$p(round_i) = P(n_i), \quad r = i.$$

$$p(round_i) = P(n_i) \prod_{j=r}^{i-1} (1 - P(n_j)), \quad r < i.$$

The probability of matching our target input message of round r to an output of round i, O_i, is (note that it is uniformly distributed over all outputs of round i, s_i):

$$p(O_i) = \frac{P(n_i)}{s_i}, \quad r = i.$$

$$p(O_i) = \frac{P(n_i) \prod_{j=r}^{i-1} (1 - P(n_j))}{s_i}, \quad r < i.$$

This result only makes sense if $s_i > 0$. Otherwise, $p(O_i) = 0$, and this term should not count in the computation of the entropy. The recipient anonymity of the input, assuming that the probability of it still being in the mix is negligible after round R, is:

$$H_r = -\sum_{i=r}^{R} s_i \cdot p(O_i) \log(p(O_i)) \ . \tag{1}$$

6 Sender anonymity without dummy traffic

In order to compute the sender anonymity, we want to obtain the effective size of the anonymity set of senders for a message output by the mix. Therefore, we compute the entropy of the probability distribution that relates an outgoing message of the mix (the one for which we want to know the anonymity set size) with all the possible inputs.

The notation we need for this section, in addition to the one presented previously, is:

- $p(I_i)$: probability of linking the chosen output with an input I that arrived to the mix in round i.
- H_s: effective sender anonymity set size. Also *sender anonymity*.

Given that the mix treats all messages in the same way, the probability for an input to correspond to the chosen output depends on the round in which the input arrived to the mix. If a message arrived in the current round r, it is certain that it is in the pool. Therefore, the probability is uniformly distributed among all the messages contained in the mix:

$$p(I_r) = \frac{1}{n_r} \ .$$

For the messages that have arrived in previous rounds, we need to take into account that they might have already been sent by the mix. Therefore, we need to multiply the previous result by the probability of that input still being inside the mix. If the message arrived in round i, the probability of staying each round is $1 - P(n_j)$. Taking into account that the decisions of different rounds are independent, the probability of the chosen output corresponding to an input of round i is:

$$p(I_i) = \frac{1}{n_r} \prod_{j=i}^{r-1} (1 - P(n_j)), \quad i < r \ .$$

Note that the result only makes sense if the number of inputs of the round we are considering is greater that zero, otherwise $p(I_i) = 0$, and this term should not be taken into account when computing the entropy. The measure of the sender effective anonymity set size, given by the entropy, is:

$$H_s = -\sum_{i=1}^{r} a_i \cdot p(I_i) \log(p(I_i)) \ . \tag{2}$$

Note that we start at round 1 because we assume that the attacker has been permanently observing the system. From a practical point of view, if a program to measure the anonymity is embedded in the mix to evaluate the anonymity performance, this program will be started at the same time as the mix, and will also "know" the whole history of it.

7 Sender anonymity with dummy traffic

In this section we discuss the sender anonymity metric when dummy traffic is generated by the mix. We consider two scenarios: dummies inserted at the output and in the pool. We reason that the intuitive way of computing this anonymity results in a metric that does not reflect the actual increase in the anonymity of the users.

7.1 Dummies inserted at the output

We encounter the first limitation of the metric when trying to measure the sender anonymity in a setting in which the mix is producing dummies.

In order to compute the sender anonymity provided by the mix when dummy traffic is being inserted at the output link, we would first choose an output, and then compute the probability of this message being one of the inputs or one of the dummies. There is a conceptual difference between these two cases: if the output is a real message, we want to know *which one*; if it is a dummy, we do not really care whether it is "dummy number 1" or "dummy number 7": the fact of the message being a dummy contains only one bit of information (dummy/no dummy). We show that treating the two cases analogously would lead to a metric that is not meaningful in terms of anonymity.

Let us consider a distribution of probabilities p_i that relates the chosen output with every possible input I_i when no dummies are generated by the mix. The entropy of this distribution is H_s. If the mix adds d_k messages to every output round, then the new probability distribution is:

- Probability of being a dummy: $p_d = d_k/s_k$.
- Probability of being input I_i: $(1 - p_d) \cdot p_i$

The entropy of the new distribution is:

$$H = -p_d \log_2(p_d) - \sum_i (1 - p_d) \cdot p_i \log_2((1 - p_d) \cdot p_i) \ .$$

$$H = -p_d \log_2(p_d) - (1 - p_d) \log_2(1 - p_d) + (1 - p_d) \cdot H_s \ .$$

From the formula, we observe that for high values of H_s and p_d, the value of the new entropy H (with dummies) may be lower than H_s (entropy with no dummies).

The decrease in the entropy is consistent with the concept associated with it: the *uncertainty*. If $p_d >> 1 - p_d$, the attacker has little uncertainty about the output, he may guess that it is a dummy and he will be right with probability p_d. Nevertheless, the attacker is not gaining much with this guess because the uncertainty about the inputs that corresponds to real outputs stays the same.

We should conclude that it is not straighforward to use the metric H to compute the sender anonymity of a mix with dummy traffic. In order to get meaningful results, we should assume that the attacker chooses a real message, and never a dummy. As complementary information about the chances of the attacker of choosing a real message at the output of a mix, we suggest to provide, together with the metric H_s, the probability of success choosing a real message, $1 - p_d$.

On the other hand, we should note that the incoming dummies that are discarded by the mix do contribute to the sender anonymity.

7.2 Dummies inserted in the pool

The same problem pointed out in the previous section about the relevance of the metric applies to this scenario, hence the same solution is suggested. We propose as metric the entropy conditioned to the event that a real message is

chosen, together with the probability of choosing a real message, $1 - p_d$ - as in the previous case.

The main difference witht he previous case is that for binomial mixes the number of dummies flushed by the mix follows a binomial distribution with respect to the number of dummies contained in the pool. The average number of dummies contained in the pool at round r is:

$$D_r = d_r + \sum_{i=1}^{r-1} d_i \prod_{j=i}^{r-1} (1 - P(n_j)) \ .$$

The proportion of dummies at the output is, on average, the same as in the pool (the dummies are selected to be sent with the same probability as real messages). The probability of selecting a real message at the output is: $1 - p_d = 1 - D_r/n_r$.

Note that the entropy in this scenario must be computed taking into account the actual value of $P(n)$ (where n includes the dummies). The value is higher than in the case in which dummies are inserted at the output. Therefore, the mix may provide less anonymity and less delay. Note that the value of the function $P(n)$ depends not only on the number of real messages contained in the pool, but also on the number of dummies. This implies that n_k will be bigger that in the other analyzed cases. $P(n)$ is a function that grows with n (a function that decreases with n would not make sense: the mix would send less messages as the traffic increases). From the expression of the entropy, we can conclude that for the same traffic load, the anonymity and the delay decrease when this policy is used instead of inserting the dummies at the output (note that higher values of $P(n)$ provide less anonymity and less delay). Eventually, we could reach a situation in which a real message is only mixed with dummies. Note that if the function $P(n)$ does not increase its value ($P(n)$ may reach a maximum value), the anonymity would not be affected.

8 Recipient anonymity with dummy traffic

A similar problem arises for the case of recipient anonymity as for sender anonymity. In this case, we must assume that the attacker choses to trace a real input. This is a reasonable assumption when the message comes from the user. But in certain circumstances, the attacker may want to trace a message that comes from another mix (trying to find the path of the target message in the network). In this case, the attacker may choose a message that is actually a dummy that will be discarded by the mix. It does not seem easy to model the dummy traffic that arrives to a mix for being discarded, given that it depends on the whole network and the path of the dummy.

In order to effectively apply the anonymity metric, we must assume that the attacker computes the recipient anonymity for a message that will not be discarded by the mix (that is, a message that matches an output). Analogously to the case of sender anonymity, we may provide as complementary information to the recipient anonymity, the probability of choosing an input message that is not discarded by the mix.

In this section we discuss the impact of the dummy traffic created by the mix on the recipient anonymity. We show that a simple extension of the metric allows us to take into account dummy traffic generated by this mix (the input dummy traffic getting to the mix cannot be considered). We compare the two possible ways of inserting dummies: at the output and in the pool. The number of dummies inserted at round k is d_k. The number of dummies inserted follows a distribution $\Pr(d_k = d)$. We make abstraction of this distribution.

8.1 Dummies inserted at the output

The mix inserts d_k messages at the output link in round k. The recipient anonymity when dummy traffic is being inserted at the output of the mix is computed using (1). The only difference in this case is that s_k has a component of real messages, m_k, and another one of dummy messages, d_k ($s_k = m_k + d_k$). Therefore, the impact of the dummy traffic is equivalent to an increase in the traffic load.

This simple result is consistent with the fact that real messages which are not the one we want to trace act as cover traffic for the target message, just as dummy messages do. Whenever there is at least one real message in the output of a round, the probabilities of matching our target input message are distributed over the messages output by the mix in that round.

Nevertheless, it is important to note that if m_k and d_k are known by the attacker (deterministic mix or deterministic dummy policy), the rounds in which $m_k = 0$ (only dummy messages sent) can be discarded by the attacker. These dummy messages do not increase the recipient anonymity provided by the mix. This is not the case when the attacker has uncertainty about d_k and m_k (binomial mix with random dummy policy); therefore he has to take into account dummies sent in rounds in which no real message is flushed.

We can conclude that binomial mixes with random dummy policy offer more anonymity when the traffic is low (in particular, when $m_k = 0$), because the uncertainty of the attacker about the existence of real messages in the output increases the recipient anonymity: messages of rounds that would be discarded by the attacker in a deterministic mix cannot be discarded in a binomial mix.

8.2 Dummies inserted in the pool

The mix inserts in the pool d_k dummies in round k. The recipient anonymity provided by a mix implementing this dummy policy is computed using (1). The difference in this case is that the value of the function $P(n)$ depends not only on the number of real messages contained in the pool, but also on the number of dummies, with the same consequences on the anonymity as mentioned in Sect. 7.2.

9 Recipient anonymity under $n-1$ attack

The $n-1$ or *blending* attack (analyzed in detail by Serjantov *et al.* in [SDS02]) is a method to trace a message going through a mix. The goal of this attack is to identify the recipient of a message (the attack only affects recipient anonymity, not sender anonymity). In order to deploy an $n-1$ attack, the attacker fills the mix with his own messages and the target message (he must be able to delay the other incoming messages). Assuming that the attacker can recognize his messages at the output, then he is able to trace the target message. In this attack model, the adversary is able to delay messages and to generate large numbers of messages from distributed sources (so that the flooding of the mix cannot be distinguished from a high traffic load).

If no dummy traffic is being generated by the mix, then the attacker can successfully trace the target (with probability 1 for a deterministic mix and with arbitrarily high probability for a binomial mix).

9.1 Deterministic mix with dummy traffic inserted at the output

In this case, the attacker knows d_k and m_k. Therefore, he knows when the target message is being sent by the mix (it is the round in which the number of unknown messages sent is $d_k + 1$). The anonymity will be that provided by the dummies in the round in which the target is flushed (round i):

$$H_r = -\sum_{j=1}^{d_i+1} \frac{1}{d_i+1} \log_2\left(\frac{1}{d_i+1}\right) = \log_2(d_i+1) \ .$$

Note that although the attacker can detect the round in which the target message is flushed, he still cannot distinguish between the target message and the dummies.

9.2 Binomial mix with random dummy traffic inserted at the output

In this case, the attacker cannot observe in which round the message is flushed, because he does not know d_k and m_k, and he cannot distinguish between the dummies and the target message. We assume that after round R the probability of the target message being inside the mix is negligible.

The mix flushes s_k messages per round. The attacker can recognize m_k messages. He does not know whether the $s_k - m_k$ remaining messages are just dummies or if the target is among them.

The attacker fills the mix with his own messages and lets the target in at round r. From that round on, the probability of every unknown output of round i of being the target is:

$$p(O_i) = \frac{P(n_i)}{s_i - m_i}, \quad r = i.$$

$$p(O_i) = \frac{P(n_i)}{s_i - m_i} \prod_{j=r}^{i-1} (1 - P(n_j)), \quad r < i.$$

The entropy is given by:

$$H = -\sum_{i=r}^{R} (s_i - m_i) \cdot p(O_i) \log(p(O_i)) \ .$$

This means that all the dummies sent in the rounds in which there is a probability of sending the target (this includes the rounds before and/or after the actual sending of the target) contribute to the anonymity, in contrast with the previous case, in which the round that includes the target is observable and only the dummies sent in that particular round contribute to the recipient anonymity of the message.

9.3 Dummies inserted in the pool

If the dummies are inserted in the pool, then the attacker has uncertainty about the round in which the target message is flushed. This is independent of the type of mix (deterministic or binomial) and the dummy distribution (deterministic or random dummy policy): the attacker can neither distinguish at the output between unknown real messages and dummy messages, nor know which of the messages of the pool will be selected.

The anonymity provided in this case is computed as in the case of binomial mixes with random dummy policy. The only difference is that the pool will contain more messages (n grows due to the dummies). This increases $P(n)$, unless $P(n)$ reaches at a certain point a maximum (as it is the case in some practical designs, as Mixmaster) and the attacker sends enough messages to make it reach this maximum. An increase in the result of the function $P(n)$ would help the attacker to force the target to leave the mix in fewer rounds with a high probability.

10 Conclusions and future work

We have computed the sender and recipient anonymity provided by generalized mixes. The formulas provided are compact and easy to evaluate and implement. We have indicated how to measure the sender and recipient anonymity when the mix inserts dummy traffic in the pool or at the output. Given that the intuitive extension of the metric for this scenario provides confusing results, we have clearly explained how it should be applied. We have analyzed the anonymity provided by a mix that sends dummy traffic, when it is subject to an $n - 1$ attack, and provided the equations that express this anonymity.

We summarize the main conclusions of the paper:

- The dummies generated by the mix contribute to recipient anonymity, but not to sender anonymity. The dummies discarded by the mix contribute to sender anonymity but not to recipient anonymity. Much attention must be paid when implementing this metric to nodes that generate dummy traffic.
- Binomial mixes in combination with a random dummy policy provide more anonymity than deterministic mixes (regardless the dummy policy) or binomial mixes with deterministic dummy policy.
- Inserting the dummies in the pool provides less anonymity and less latency that inserting them at the output.
- When dummies are inserted at the output, binomial mixes with a random dummy policy offer more protection against the $n-1$ attack than deterministic mixes.
- Inserting dummies in the pool protects deterministic mixes better than inserting them at the output, when an $n-1$ attack is deployed.

Some of the topics that are subject of future work are:

- Find a metric that expresses the sender and recipient anonymity provided by a mix network with dummy traffic.
- Compare the anonymity achieved with different distributions of dummy traffic. Obtain quantitative results.
- Compare the anonymity provided by pool mixes to the anonymity provided by *Stop-and-Go* mixes, with dummy traffic.

Acknowledgments

Claudia Díaz is funded by a research grant of the K.U.Leuven. This work was also partially supported by the IWT STWW project on Anonymity and Privacy in Electronic Services (APES), and by the Concerted Research Action (GOA) Mefisto-2000/06 of the Flemish Government.

The authors also want to thank Andrei Serjantov, Joris Claessens and Dries Schellekens for their comments, questions and suggestions.

References

[Cha81] David Chaum. Untraceable electronic mail, return addresses and digital pseudonyms. *Communications of the A.C.M.*, 24(2):84–88, 1981.

[Cot] L. Cottrell. Mixmaster and remailer attacks. http://www.obscura.com/ loki/remailer/remailer-essay.html.

[Dan03] George Danezis. Mix-networks with restricted routes. In *Privacy Enhacing Technologies*, LNCS, Dresden, Germany, April 2003.

[DP04] Claudia Diaz and Bart Preneel. Taxonomy of mixes and dummy traffic. In *Accepted submission at I-NetSec04: 3rd Working Conference on Privacy and Anonymity in Networked and Distributed Systems.* Kluwer academic publishers, August 2004.

[DS03a] George Danezis and Len Sassaman. Heartbeat traffic to counter (n-1) attacks. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2003)*, Washington, DC, USA, October 2003.

[DS03b] Claudia Diaz and Andrei Serjantov. Generalising mixes. In *Privacy Enhacing Technologies*, LNCS, Dresden, Germany, April 2003.

[DSCP02] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In *Privacy Enhancing Technologies*, April 2002.

[DSD04] Claudia Diaz, Len Sassaman, and Evelyne Dewitte. Comparison between two practical mix designs. Technical report, K.U.Leuven, 2004. Submitted to ESORICS 2004.

[Jer00] Anja Jerichow. Generalisation and security improvement of mix-mediated anonymous communication. Ph.D. thesis, Technischen Universitat Dresden, 2000.

[KEB98] D. Kesdogan, J. Egner, and R. Buschkes. Stop-and-go-mixes providing probabilistic anonymity in an open system. In *Proceedings of the International Information Hiding Workshop*, 1998.

[MC00] Ulf Moeller and Lance Cottrell. *Mixmaster Protocol Version 3*, 2000. http://www.eskimo.com/ rowdenw/crypt/Mix/draft-moeller-v3-01.txt.

[PK00] Andreas Pfitzmann and Marit Kohntopp. Anonymity, unobservability and pseudonymity — a proposal for terminology. In *Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on the Design Issues in Anonymity and Observability*, pages 1–9, July 2000.

[SD02] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *Privacy Enhacing Technologies*, LNCS, San Francisco, CA, April 2002.

[SDS02] Andrei Serjantov, Roger Dingledine, and Paul Syverson. From a trickle to a flood: Active attacks on several mix types. In F. Petitcolas, editor, *Information Hiding Workshop*, October 2002.

[SN03] Andrei Serjantov and Richard E. Newman. On the anonymity of timed pool mixes. In *Workshop on Privacy and Anonymity in Networked and Distributed Sy stems (18th IFIP International Information Security Conference)*, Athens, Greece, May 2003.