TAXONOMY OF MIXES AND DUMMY TRAFFIC

Claudia Diaz
K.U.Leuven Dept. Electrical Engineering-ESAT/COSIC
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
claudia.diaz@esat.kuleuven.ac.be

Bart Preneel
K.U.Leuven Dept. Electrical Engineering-ESAT/COSIC
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
bart.preneel@esat.kuleuven.ac.be

Abstract This paper presents an analysis of mixes and dummy traffic policies, which are building blocks of anonymous services. The goal of the paper is to bring together all the issues related to the analysis and design of mix networks. We discuss continuous and pool mixes, topologies for mix networks and dummy traffic policies. We point out the advantages and disadvantages of design decisions for mixes and dummy policies. Finally, we provide a list of research problems that need further work.

Keywords: Mixes, Mix Networks, Anonymity, Dummy Traffic

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 personal information; it also means not revealing information about who is communicating with whom. Therefore, anonymity needs to be implemented at the communication and application layer in order to effectively protect the users' privacy.

Mixes are a basic building block for anonymous applications. In this paper we present an analysis of mixes and dummy traffic. We discuss all the issues that need to be taken into account when analyzing or designing mixes and dummy policies for mix networks. This paper intends to be a starting point for those who are new to the field of anonymous services as well as a support for

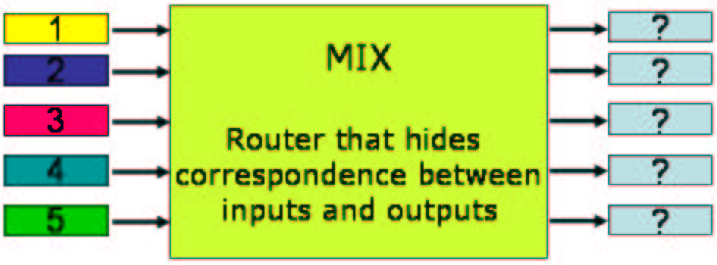


Figure 1. Mix

designers of mixes and dummy policies. We also point out the problems that remain unsolved in this field.

Road-map of the Paper. Section 2 introduces the basic concept of a mix. Section 3 presents the distinction between the two main families of mixes: continuous and pool mixes. Section 4 discusses the issues related to continuous mixes, while Section 5 analyzes pool mixes. Section 6 presents the different topologies for a mix network. Dummy traffic is presented in Section 7. Finally, Section 8 summarizes the different aspects that need to be taken into account when designing mixes and dummy policies, and Section 9 presents the conclusions and open problems.

2. What is a Mix?

Mixes were proposed by Chaum [Chaum, 1981] in 1981. The mix takes a number of input messages, and outputs them in such a way that it is infeasible to link an output to the corresponding input (or an input to the corresponding output). In order to achieve this goal, the mix changes the *appearance* (by encrypting and padding messages) and the *flow* of messages (by delaying and reordering). A picture of a mix can be seen in Figure 1.

Changing the appearance of the message (in order to provide *bitwise unlinkability*) can be achieved with the currently available cryptographic primitives, such as randomized encryption schemes and padding schemes. Also, cryptography solves the problem of the integrity checks of the messages, needed to ensure that the contents of the message have not been modified during the transmission. Therefore, these issues are not discussed in this paper.

We need to change the flow of messages in order to make the linking of an input and an output difficult for an attacker. Modifying the flow of messages is not an easy problem, especially when the delay of the messages is constrained by real-time requirements. In this paper, we give an overview on the options that have been explored in order to anonymize the flow of messages, and we discuss the advantages and disadvantages of these designs.

2.1 Anonymity Metrics for a Mix

How to measure the degree of anonymity offered to the users of a mix? An attacker may deploy passive attacks (i.e., traffic analysis [Serjantov and Sewell, 2003]) or active attacks (e.g., the *blending* or $n-1$ attack, analyzed in detail in [Serjantov et al., 2002]) in order to identify the sender (or recipient) of a message.

The attacker can typically obtain probabilistic relationships between the inputs and the outputs of a mix. Under certain conditions (for example, low traffic, or active attacks), the attacker may be able to narrow down the set of possible senders (recipients) of a message. In other cases, one of the users will appear as having a very high probability of being the sender of a particular message. In order to achieve a good level of anonymity, we should prevent the attacker from obtaining such probability distributions.

Based on the definition for anonymity proposed by Pfitzmann and Kohntopp in [Pfitzmann and Kohntopp, 2000], two information theoretic models were independently proposed by Diaz *et al.* in [Diaz et al., 2002] and by Serjantov and Danezis in [Serjantov and Danezis, 2002]. These models measure the anonymity provided by a mix towards an attacker. Note that it is essential to clearly specify the power of the attacker before applying the anonymity metric.

The anonymity is measured using the concept of *entropy* (i.e., uncertainty), taking into account the probabilistic information that an attacker is able to obtain from the system.

These metrics may be applied to measure the uncertainty of the attacker about the sender of the message, i.e., *sender anonymity*. Analogously, the uncertainty of the attacker regarding the recipient of a message, i.e., *recipient anonymity* may be computed.

One of the limitations of this metric is that the anonymity provided by a mix cannot be computed for the theoretical design, because it needs to take into account the traffic load of the mix and the attack model considered. Therefore, it must be computed either through simulation or using a real setting. This implies that many measurements need to be performed in order to have a good estimate of the degree of anonymity provided by a particular mix. The measurements should take into account different attack models, traffic loads and traffic patterns. Some pratical results have been presented by Diaz *et al.* in [Diaz et al., 2004], where two working remailers have been analyzed (Mixmaster and Reliable). The results show that Mixmaster guarantees a minimum anonymity for all messages, regardless of the traffic load.

3. Continuous or Pool Mixes?

The original Chaumian mix [Chaum, 1981] uses the following algorithm to change the flow of messages: it collects n messages and flushes them in a batch. The attacker cannot know which of the n outputs matches a particular input and vice versa. This idea is the basis of *batching mixes*, also called *pool mixes*: a set of messages is collected by the mix and flushed when a certain condition is fulfilled. An analysis of these mixes can be found in Section 5.

A different mix concept was proposed by Kesdogan *et al.* in [Kesdogan et al., 1998]. In this design, the messages are delayed a certain amount of time, and then sent by the mix. The delay of each message is independent from the traffic load. These mixes are discussed in Section 4.

4. Continuous Mixes

The idea of continuous mixes (also called *Stop-and-Go* mixes) was first proposed by Kesdogan *et al.* [Kesdogan et al., 1998]. In this design, the users generate a random delay from an exponential distribution, and add this delay to the headers of the message. The mix holds the message for the specified delay and then forwards it. The messages are reordered by the randomness of the delay distribution. This mix sends messages continuously: every time a message has been kept for the delay time, it is sent by the mix.

4.1 Reordering Technique

In Kesdogan's original idea, the delay is chosen by the user from an exponential distribution. The exponential distribution has the advantage of being memoryless, but other distributions, such as the uniform distribution (in which the variance of the delay can be larger), may also be taken into account. A thorough study must be carried out in order to find out which design provides the best anonymity properties for the expected working context of the mix (traffic load, traffic pattern, and delay constraints). Nevertheless, Danezis shows in [Danezis, 2004] that the exponential distribution is optimal for continuous mixes.

4.2 Anonymity

Kesdogan *et al.* provide an anonymity study for the *Stop-and-Go* mix in [Kesdogan et al., 1998]. These calculations assume that the incoming traffic pattern can be approximated by a Poisson process. Real traffic arriving to a mix node in a network has been analyzed in [Diaz et al., 2004], and it has been found that the mix incoming traffic pattern is not Poisson and that it cannot be modelled by any known distribution, given that it is very unstable and impredictable.

4.3 Strengths and Weaknesses of the Design

The main advantage of this system is that the delay does not depend on the traffic that arrives to the mix. This means that tight delay constraints can be implemented by this mix, regardless of the current load of the mix (which may be useful for applications in which a small delay is more important than providing a high level of anonymity, such as web browsing applications).

Moreover, when the message is routed through a mix network (see Section 6), the user can choose the amount of time it will take to the message to arrive to every mix on the path (and to the recipient), since he is who chooses the delays of his message at each mix.

On the other hand, the anonymity provided to the users may go to low levels if the number of users decreases during a certain period of time. We must not forget that there is always a tradeoff anonymity / delay, and if we bound the delay we may drop to low levels of anonymity under certain conditions (in this case, low traffic conditions).

This design may be appropriate for systems with stable incoming traffic patterns, in which the anonymity is guaranteed by a (more or less) constant traffic rate. Systems with variable number of users and with changing traffic conditions risk to result in low levels of anonymity during quiet traffic periods, as it is shown in [Diaz et al., 2004].

These mixes are also vulnerable to *blending* or $n - 1$ attacks [Serjantov et al., 2002]. This active attack is deployed by an attacker who is able to delay the messages going to the mix. The attacker selects a *target* message he wants to trace, and delays all the other messages. In a continuous mix, this would result in the attacker being able to trace the target message, given that (with an arbitrarily high probability) the attacker can succeed in making the message going through the mix when it does not contain any other messages (the message is not *mixed*). This attack can be prevented, or at least detected, using additional mechanisms. Kesdogan proposes adding a timestamp to the messages (note that the user knows the expected time of arrival of the message to every mix); the mixes discard all messages that contain an old timestamp. Nevertheless, this technique may help detecting a *blending* attack, but it does not prevent it.

Dummy traffic (Section 7) can also be used both to prevent and to detect *blending* attacks. See Section 7.3 to find a description on how dummy traffic can be used to detect and react when a mix is subject to active attacks.

5. Pool Mixes

Pool mixes process the messages in batches. They collect messages for some time, place them in the pool (memory of the mix), and select them for flushing (in random order) when the flushing condition is fulfilled. The aspects that we should take into account when designing and analyzing a pool mix are the *flushing condition* and the *pool selection algorithm*.

Flushing condition. We can distinguish two types of mixes according to the flushing condition: *timed mixes* send messages every fixed internal time, called *timeout*. *Threshold mixes* send messages when they have collected a certain amount of messages, called the *threshold*. Some mix designs, such as Mixmaster [Mѳller et al., 2003], combine the two mechanisms: they flush when the *timeout* expires only if the *threshold* has been reached. The cycle of collecting and flushing messages is called a *round*.

So far, the mixes that have been implemented have a *fixed* timeout or threshold. It would be interesting to study the properties of mixes that choose the threshold or the timeout from a *random* distribution.

Pool selection algorithm. The performance of a pool mix (in terms of delay and anonymity) is mainly determined by the pool selection algorithm. In Chaum's design, the mix flushes all the messages it contains. Later, the concept of *pool* was added to the mix, extending the original mix to keep a number of messages (instead of flushing all of them). In the first stage, the proposals of mixes keep a fixed number of messages in the pool. Later on, mixes that kept a variable number of messages were designed (e.g., Mixmaster).

Pool algorithms enhance the anonymity (compared to Chaum's mix) by extending the anonymity set size to, potentially, an infinite number of users. Nevertheless, it should be noted that the probability distributions obtained by an attacker trying to trace a message will not be uniform for all senders (or recipients) of messages.

The parameters that should be taken into account when designing a pool selection algorithm are the number of messages kept in the pool (which can be fixed or variable, e.g., percentage of the total number of messages at the time of flushing); and the number of messages sent (which can also be fixed or variable).

Section 8 gives a summary of the relevant parameters in the design of a mix.

5.1 The Generalised Mix Model

The Generalized Mix Model was proposed by Diaz and Serjantov in [Diaz and Serjantov, 2003]. This model can express pool mixes by abstracting of the flushing condition and representing in the graph the pool selection algorithm. The mix is represented at the time of flushing; it shows the percentage of messages contained in the mix that are sent in a round, as a function $P(n)$ of the total number of messages in the mix.

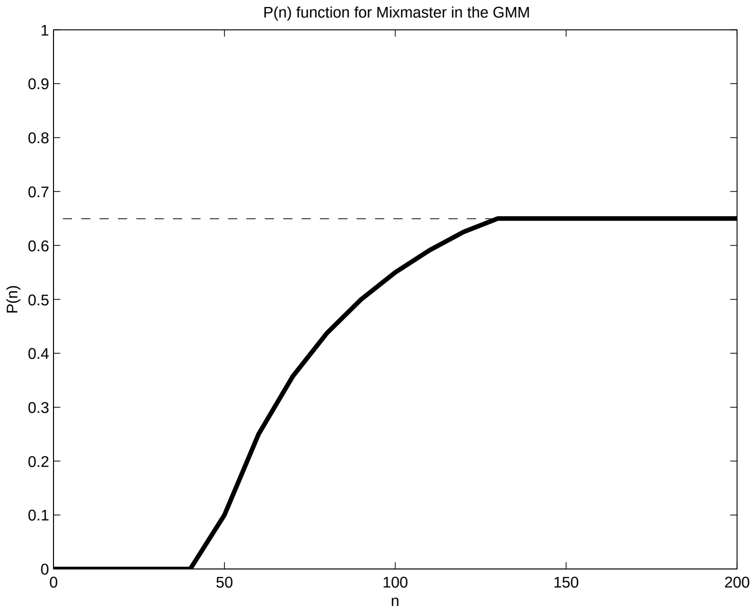


Figure 2. Representation of a Cottrell mix in the Generalised Mix Model

A representation of the flushing algorithm of Mixmaster (designed by Cottrell) is shown in Figure 2. The algorithm is as follows:

- If $0 < n < 45$ do not send any message (i.e., $P(n) = 0$)
- If $45 < n < 129$ send $n - 45$ messages (i.e., $P(n) = 1 - 45/n$)
- If $n > 129$ send $0.65 * n$ messages (i.e., $P(n) = 0.65$)

The function that represents the mix in the Generalised Mix Model is very useful to implement anonymity metrics, because it contains all the mix-related data needed to compute the anonymity offered by the mix. It also provides an intuitive idea on the performance of the mix under different traffic loads, which is closely related to this function: high values of the function favor low delays over high levels of anonymity, while low values of the function enhance the anonymity at the cost of larger delays. The model allows for the design of mixes that implement complex pool selection algorithms in an easy and intuitive way.

5.2 Deterministic or Binomial?

The mix function $P(n)$ denotes the probability of sending a message, given that there are n messages in the mix. 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 n of messages in the pool is known, the number s of messages sent is determined, and vice versa.

Binomial mixes. Binomial mixes were introduced in [Diaz and Serjantov, 2003]. In these mixes, an independent decision is taken for every message in the pool. A biased coin (its bias is equal to 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):

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

The probability of having n messages in a pool of maximum size N_{max}, given that the mix sends s messages is [Diaz and Serjantov, 2003]:

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

This probabilistic relationship has the following implications: as it was shown in [Diaz and Serjantov, 2003], 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 pointed out by Diaz and Preneel in [Diaz and Preneel, 2004].

5.3 Strengths and Weaknesses of the Design

The main advantage of pool mixes is their ability to adapt to fluctuations in the traffic load. If a good mix function is selected, the mix can compensate a low traffic load by introducing more delay, in order to keep a good anonymity level. These mixes are ideal for applications that do not have tight delay constraints, such as anonymous email.

On the other hand, the delay introduced by a pool mix is not predictable by the sender of the message. This becomes worse when the message is routed through a mix network. Therefore, pool mixes are not appropriate for real-time applications. A comparison between practical pool and continous mixes can be found in [Diaz et al., 2004]

Regarding the vulnerability to *blending* or $n-1$ attacks, the success of the attacker strongly depends on the details of the mix design. The attacker needs to be able to delay messages going to the mix and also to generate messages that are accepted by the mix (which was not required to attack continuous mixes). We point out the following cases:

- *Threshold mixes* are more vulnerable than *timed mixes* (or mixes that combine the *threshold* with a *timeout*), because the attacker can succeed in emptying the mix from valid unknown messages in a short time by simply flooding the mix with his own messages.

- Mixes that do not have a pool (i.e., they do not keep messages from one round to another) are extremely vulnerable to *n-1* attacks. The attacker can be sure to succeed in tracing the target message, and he only needs to attack the mix for one round.

- Deterministic pool mixes require a stronger effort from the attacker, who needs to attack the mix for several rounds in order to trace a single message. Nevertheless, a powerful attacker is able to successfully trace a message when it goes through one of this mixes.

- Binomial pool mixes are more robust than deterministic pool mixes under *n-1* attacks. The success of the attacker becomes only probabilistic, and the effort required to the attacker grows.

Dummy traffic policies, discussed in Section 7, help preventing and detecting *blending* or *n-1* attacks.

6. Mix Networks

In order to increase the anonymity of a mix system, mixes are usually combined in a mix network. This way, the fact that some mixes are corrupted or controlled by an attacker does not break the anonymity of the users (the anonymity of a message is guaranteed even if only one of the mixes in the path of the message is honest). Also, the reliability of the system is improved, because the failure of a mix does not lead to a denial of service.

6.1 Cascades, Free Route Networks and Restricted Route Networks

The two classical topologies of mix network are cascades and free route networks. In a cascade, the possible paths that a message can follow are predefined (it can be one or more). This is the approach followed by [JAP Anonymity & Privacy,]. In a free route network, users select freely their own path, which may be different for every message. Onion Routing [Goldschlag et al., 1996] and Mixmaster [Mɵller et al., 2003] are examples of free route mix networks. The advantages and disadvantages of these two topologies have been pointed out by Berthold *et al.* in [Berthold et al., 2000].

More recently, Danezis proposed in [Danezis, 2003] a mix network topology that is somehow in between the two classical designs. In this model, every mix node communicates with a few neighboring others. The goal of this idea is to combine the advantages of cascades and free route networks and overcome the disadvantages.

6.2 Inter-Mix detours

This technique has been proposed in [Gʉlcʉ and Tsudik, 1996]. It consists of giving to the mixes the ability to re-encrypt a message at any point of the network and send it through a detour before it goes back to the original path. This increases the latency of the network, but enhances the anonymity of the messages. Nevertheless, we do not have any tools yet that evaluate the effectiveness of this technique and the optimal values for the following parameters:

- Probablity of sending a message through a detour.
- Route length of the detour.
- Route selection of the detour.

7. Dummy Traffic

A dummy message is a *fake* message introduced in a mix network in order to make it more difficult for an attacker to deploy passive and active attacks. Dummy messages are normally generated by the mixes (although users may also generate dummies, which increases the anonymity level of the mix network and prevents end-to-end intersection attacks [Berthold and Langos, 2002]); they have as destination another mix, instead of a real recipient. Dai proposed the Pipenet system [Dai, 1996] a system in which the traffic is constant: the links between mixes are padded with dummy messages whenever the real traffic is not enough to fill them. This system provides not only anonymity, but also unobservability, since an observer of the network cannot tell whether there are real messages traveling in the network or not. Unfortunately, the system is not practical due to the enormous amount of resources it needs.

The generation and transmission of dummy traffic has a cost, and it is therefore very important to find the right balance on the number of dummies that should be created in a mix network. The rest of this section studies the possible choices we can make when designing a dummy policy.

7.1 Generation of Dummies

The first question that arises when designing a dummy traffic policy is whether the dummies generated should depend on the incoming traffic or not. Generating dummies depending on the traffic load may make a more efficient use of the resources, but this dependency can be exploited by an active attacker to maximize the effectiveness of his attack by generating his own messages in such a way that he minimizes the number of dummies generated by the mix. Therefore, dummy traffic policies that are independent from the traffic load seem to be more secure.

One of the issues that needs to be decided is the average number of dummies we want to generate (for pool mixes we will choose an average number of dummies per round, while in continuous mixes we will generate dummies per fixed time unit). These dummies can be generated following a deterministic or random distribution. Random distributions increase the uncertainty of the attacker, specially when combined with binomial mixes, as pointed out in [Diaz and Preneel, 2004].

Continuous mixes. These mixes may generate a certain number of dummies every period of time, selecting their delay (amount of time they are kept in the mix from their generation until the moment in which they are sent) from a random distribution. This is the approach followed by *Reliable*, one of the mixes that composes the Mixmaster network.

Other dummy policies may be explored, for example, the mix could keep always one dummy inside, and generate a new one (with its corresponding delay) when the dummy is sent. Another policy would be that the mix decides every certain amount of time on whether to generate a dummy or not.

Pool mixes. The design of dummy policies for pool mixes implies making decisions on the following issues:

- The dependency on the traffic load.
- The average number of dummies generated per round.
- The distribution followed to select the number of dummies in a particular round (binomial, uniform, geometrical, etc.).

- Whether the dummies are inserted in the pool or at the output.
- Route length and selection of path for the dummies.

Insertion in the Pool. With this technique, the mix inserts the dummies it generates for a round in the pool. These dummies are treated as real messages by the mix after being placed in the pool.

Insertion at the Output. If the mix is to insert the dummies at the output, then it adds the dummies to the batch of real messages taken from the pool. The mix does not modify the number of messages contained in the pool.

The advantages and disadvantages of these two dummy insertion options have been discussed in [Diaz and Preneel, 2004]. Here, we summarize the conclusions presented in [Diaz and Preneel, 2004]:

- Inserting the dummies in the pool provides less anonymity and less delay 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.

7.2 Route Length and Selection of Path

Dummy messages, just like real messages, travel in the mix network going through a number of mixes. The route length of the dummy determines the number of mixes a dummy is going through. Regarding this issue, we should decide on the average number of mixes in the path of the dummy and on the distribution of this route length. Random distributions increase the uncertainty of the attacker with respect to a deterministic distribution (i.e., fixed number of mixes in the path) when the attacker wants to find out whether a message is a dummy or not.

Normally the path of a dummy is selected randomly among the mixes of the network. The last mix in the path of the dummy can be the mix that actually generated it, preventing this way that corrupted mixes can help the attacker (when they are the last in the path of the dummy) providing the information on which messages were dummies. Note that intermediate mixes (i.e., except for the first and last in the path of the dummy) cannot distinguish dummy messages from real messages.

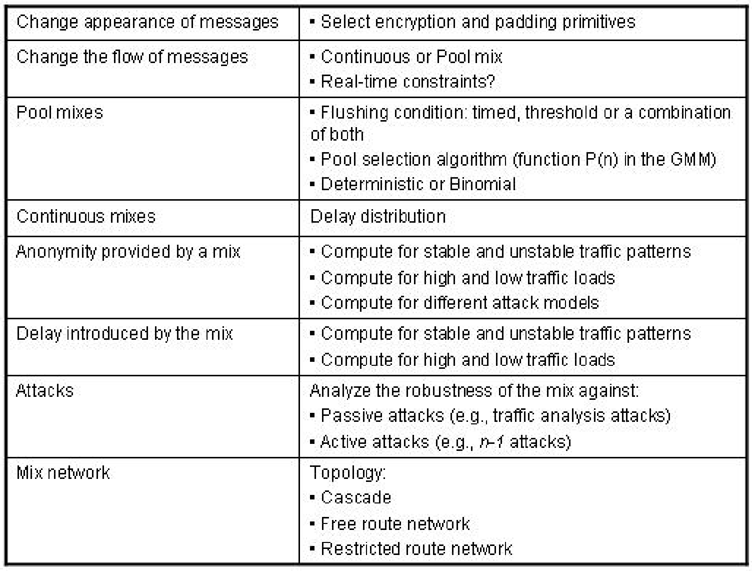
Note that, in order to increase the anonymity provided by the mix, the mix should maximize the number of possible destinations for every message, meaning that the mix should check if it is sending messages to all the possible neighbours. If it is not, then it should generate some extra dummies to send to those mixes. This way, an attacker wanting to trace a message will have to follow more possible paths.

7.3 RGB Dummy Policies

This dummy policy was proposed by Danezis and Sassaman in [Danezis and Sassaman, 2003]. The goal is to detect and counter active attacks (such as the *n-1* attack). The basic idea of this dummy policy is that the mix generates dummies that after being routed through the network are sent back to the mix that generated them. If the mix receives less dummy messages than expected, it may assume that it is subject to an $n - 1$ attack, and it reacts by stopping its functioning until the attack is no longer being deployed.

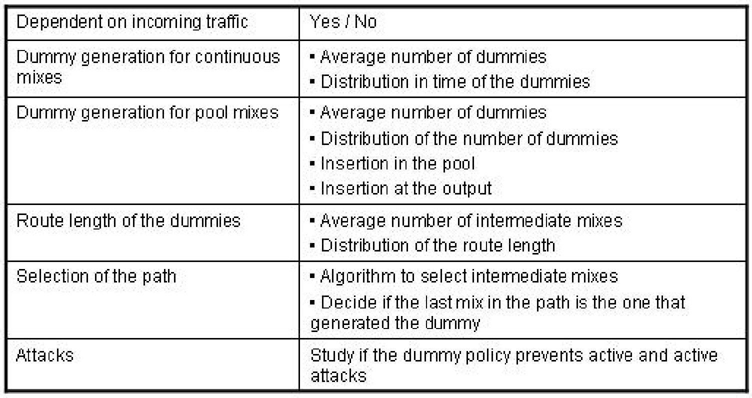
8. Summary

In this section we present a summary of the different aspects that have to be taken into account when designing mixes and mix networks, as shown in Figure 3 and a summary of the parameters of a dummy policy, in Figure 4.



Change appearance of messages	▪ Select encryption and padding primitives
Change the flow of messages	▪ Continuous or Pool mix ▪ Real-time constraints?
Pool mixes	▪ Flushing condition: timed, threshold or a combination of both ▪ Pool selection algorithm (function P(n) in the GMM) ▪ Deterministic or Binomial
Continuous mixes	Delay distribution
Anonymity provided by a mix	▪ Compute for stable and unstable traffic patterns ▪ Compute for high and low traffic loads ▪ Compute for different attack models
Delay introduced by the mix	▪ Compute for stable and unstable traffic patterns ▪ Compute for high and low traffic loads
Attacks	Analyze the robustness of the mix against: ▪ Passive attacks (e.g., traffic analysis attacks) ▪ Active attacks (e.g., *n-1* attacks)
Mix network	Topology: ▪ Cascade ▪ Free route network ▪ Restricted route network

Figure 3. Parameters of mixes and mix networks



Dependent on incoming traffic	Yes / No
Dummy generation for continuous mixes	▪ Average number of dummies ▪ Distribution in time of the dummies
Dummy generation for pool mixes	▪ Average number of dummies ▪ Distribution of the number of dummies ▪ Insertion in the pool ▪ Insertion at the output
Route length of the dummies	▪ Average number of intermediate mixes ▪ Distribution of the route length
Selection of the path	▪ Algorithm to select intermediate mixes ▪ Decide if the last mix in the path is the one that generated the dummy
Attacks	Study if the dummy policy prevents active and active attacks

Figure 4. Parameters of a dummy policy

9. Conclusions and Open Problems

In this paper we have presented a thorough analysis of the parameters of mixes and dummy traffic policies, distinguishing between continuous and pool mixes. We have discussed the advantages and disadvantages of different design options. We have introduced anonymity metrics and mix network topologies.

Some of the problems that remain unsolved in this field are:

- The current anonymity metrics can measure the anonymity provided by a mix in a simulation or in a working setting, but we do not have yet theoretical tools that allow us to know the anonymity properties of the mix during the design phase. Nevertheless, we may use simulations in order to see the anonymity that the mix can provide.

- The anonymity metrics are very useful to measure the anonymity provided by a single mix, but they fail to measure the end-to-end anonymity provided by a mix network. An extension to the metric needs to be found in order to have practical tools to measure the anonymity provided by a mix network.

- Much research need to be done in order to solve many dummy traffic related problems. We do not know yet which is the most appropriate distribution for the generation of dummies, the route length they should have in order to optimize the cost/anonymity relationship, whether they should be inserted in the pool of the mix or at the output, whether dummy traffic should depend on the real traffic traveling in the network or not, and how this dependency should be.

- Different mix designs need to be compared in order to find the best performing mixes.

Acknowledgments

Claudia Diaz 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.

References

[Berthold and Langos, 2002] Berthold, Oliver and Langos, Heinrich (2002). Dummy traffic against long term intersection attacks. In Dingledine, Roger and Syverson, Paul, editors, *Proceedings of Privacy Enhancing Technologies workshop (PET 2002)*. Springer-Verlag, LNCS 2482.

[Berthold et al., 2000] Berthold, Oliver, Pfitzmann, Andreas, and Standtke, Ronny (2000). The disadvantages of free MIX routes and how to overcome them. In Federrath, H., editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 30–45. Springer-Verlag, LNCS 2009.

[Chaum, 1981] Chaum, David (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2).

[Dai, 1996] Dai, Wei (1996). Pipenet 1.1. Usenet post.

[Danezis, 2003] Danezis, George (2003). Mix-networks with restricted routes. In Dingledine, Roger, editor, *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*. Springer-Verlag, LNCS 2760.

[Danezis, 2004] Danezis, George (2004). The traffic analysis of continuous-time mixes. In *Accepted submission at PET2004*.

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

[Diaz and Preneel, 2004] Diaz, Claudia and Preneel, Bart (2004). Reasoning about the anonymity provided by pool mixes that generate dummy traffic. In *Accepted submission at IH2004*.

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

[Diaz and Serjantov, 2003] Diaz, Claudia and Serjantov, Andrei (2003). Generalising mixes. In Dingledine, Roger, editor, *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*. Springer-Verlag, LNCS 2760.

[Diaz et al., 2002] Diaz, Claudia, Seys, Stefaan, Claessens, Joris, and Preneel, Bart (2002). Towards measuring anonymity. In Dingledine, Roger and Syverson, Paul, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482.

[Goldschlag et al., 1996] Goldschlag, David M., Reed, Michael G., and Syverson, Paul F. (1996). Hiding Routing Information. In Anderson, R., editor, *Proceedings of Information Hiding: First International Workshop*, pages 137–150. Springer-Verlag, LNCS 1174.

[Gʉlcʉ and Tsudik, 1996] Gʉlcʉ, Ceki and Tsudik, Gene (1996). Mixing E-mail with Babel. In *Proceedings of the Network and Distributed Security Symposium - NDSS ’96*, pages 2–16. IEEE.

[JAP Anonymity & Privacy,] JAP Anonymity & Privacy. http://anon.inf.tu-dresden.de/.

[Kesdogan et al., 1998] Kesdogan, Dogan, Egner, Jan, and Bʉschkes, Roland (1998). Stop-and-go MIXes: Providing probabilistic anonymity in an open system. In *Proceedings of Information Hiding Workshop (IH 1998)*. Springer-Verlag, LNCS 1525.

[Mɵller et al., 2003] Mɵller, Ulf, Cottrell, Lance, Palfrader, Peter, and Sassaman, Len (2003). Mixmaster Protocol — Version 2. Draft.

[Pfitzmann and Kohntopp, 2000] Pfitzmann, Andreas and Kohntopp, Marit (2000). Anonymity, unobservability and pseudonymity — a proposal for terminology. In Federrath, H., editor, *Designing Privacy Enhancing Technologies: Proceedings of the International Workshop on the Design Issues in Anonymity and Observability*, pages 1–9. Springer-Verlag, LNCS 2009.

[Serjantov and Danezis, 2002] Serjantov, Andrei and Danezis, George (2002). Towards an information theoretic metric for anonymity. In Dingledine, Roger and Syverson, Paul, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482.

[Serjantov et al., 2002] Serjantov, Andrei, Dingledine, Roger, and Syverson, Paul (2002). From a trickle to a flood: Active attacks on several mix types. In Petitcolas, Fabien, editor, *Proceedings of Information Hiding Workshop (IH 2002)*. Springer-Verlag, LNCS 2578.

[Serjantov and Sewell, 2003] Serjantov, Andrei and Sewell, Peter (2003). Passive attack analysis for connection-based anonymity systems. In *Proceedings of ESORICS 2003*.