Heartbeat Traffic to Counter (n-1) Attacks

Red-Green-Black Mixes

George Danezis
University of Cambridge, Computer Laboratory,
William Gates Building, 15 JJ Thomson Avenue,
Cambridge CB3 0FD, United Kingdom.
George.Danezis@cl.cam.ac.uk

Len Sassaman
Anonymizer, Inc.
5694 Mission Center Road, PMB 426
San Diego, CA 92108, USA.
rabbi@anonymizer.com

ABSTRACT

A dummy traffic strategy is described that can be implemented by mix nodes in an anonymous communication network to detect and counter active $(n-1)$ attacks and their variants. *Heartbeat* messages are sent anonymously from the mix node back to itself in order to establish its state of connectivity with the rest of the network. In case the mix is under attack, the flow of heartbeat messages is interrupted and the mix takes measures to preserve the quality of the anonymity it provides by introducing decoy messages.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms

Security

Keywords

anonymity, mix networks, flooding attacks

1. INTRODUCTION.

Mix networks have been proposed by Chaum [2] in order to provide for anonymous messaging over communication networks. A mix is a network node that hides the correspondence between its inputs and outputs, using a combination of encryption, padding, batching and delaying strategies. In order for users not to rely on a single entity to protect their anonymity, mixing can be distributed and performed by a sequence of nodes arranged in a cascade or a network.

Attacks against mix networks can be passive and aim at linking input and output messages. Passive attackers observe all the public data on the network and try to infer from them the hidden relations between senders and receivers. Active attackers try to subvert the correct functioning of the network by injecting, deleting or delaying arbitrary messages in the network. While Chaum presents an argument about the security of mixes against passive adversaries, a number of active attacks have been described over the years.

The most powerful active attack is the $(n-1)$ attack, performed by flooding a node with fake messages alongside a single message to be traced. The attacker can recognise her messages and therefore link the sender with the receiver of the single message under surveillance. This attack is active in the sense that it critically depends on the adversary's ability to inject fake messages in order to flood the honest node and delete or delay other genuine messages except the one under surveillance (in addition to the ability to observe arbitrary network links). The attack also depends on the mix's inability to detect that such an attack is taking place in order to react, either by stopping its operation or trying to confuse or deceive the attacker.

In the present work we describe a technique that mix nodes can employ to detect if they are under an active attack. The technique relies upon individual mixes being aware of their network environment and their state of connectivity with it by sending anonymous messages through the network back to themselves. We call these *heartbeat* messages, or red traffic. When a mix is under attack it cannot directly detect how much of the traffic it receives (black traffic) is genuine or simply the attackers' flooding traffic. Therefore the mix tries to estimate the amount of flooding traffic from the rate of heartbeat messages and injects dummy traffic (green traffic) in order to artificially increase the anonymity provided to honest messages.

The key intuition for understanding the properties of RGB-mixes is that the different colours of the traffic can only be observed by the mix itself. To all other mixes or attackers, traffic exiting the mix looks the same. In order to perform the $(n-1)$ attack, an attacker would need to delete or delay selected messages while simultaneously allowing the mix to receive heartbeat messages. While an attacker flooding the mix will be able to distinguish her black messages from other messages exiting the mix, the attacker is prevented from filtering out genuine traffic from heartbeat messages. Thus, the number of heartbeat messages received can be used by the mix to estimate the number of honest messages present in the mix's input.

WPES'03 October 30, 2003, Washington, DC, USA.
Copyright 2003 ACM 1-58113-776-1/03/0010 ...$5.00.

2. RELATED WORK.

Active attacks, and in particular the $(n-1)$ attack, were known in different communities working on anonymous communications [4] for a long time. In their survey of mixing strategies Serjantov *et al.* [8] assess the effectiveness of different mixing strategies against a number of active attacks. They calculate the number of rounds that it would take an attacker to be successful, and find that some mix strategies are more expensive to attack than others. On the other hand no mixing strategy provides an absolute defence since they can all be attacked in a finite amount of time or rounds. The $(n-1)$ attack (applicable primarily to threshold mixes) is generalised for other mixing strategies and called a blending attack. It is a simultaneous trickle attack, namely stopping genuine messages, and a flooding attack, that fills the mix with the attacker's messages.

In designing sg-mixes to resist $(n-1)$ attacks Kesdogan *et al.* [5] followed a different approach. They observe that the ability to realistically perform the $(n-1)$ attack relies on delaying rather than deleting messages. Therefore if messages follow a tight schedule in the network, and are dropped if they are late, an attacker would have to destroy traffic and ultimately the network would become aware of the attack. Furthermore only a fraction of the traffic could be attacked at any time. In order to provide 'real time' guarantees, they use a continuous mixing strategy based on delaying messages according to an exponential distribution. Messages contain timestamps and are delayed for as much as it is requested by the original sender. If a message misses its deadlines it is dismissed.

Mixmaster [6], the only widely deployed mix network, uses dummy traffic to foil $(n-1)$ attacks. A random number of dummy messages are included in the message pool every time an message arrives from the network. This is an effective, but quite expensive strategy since dummy messages are sent even during normal operation.

Other mix designs, such as Mixminion [3], use link encryption that makes it difficult for an attacker to recognise even her own messages in the network. This can be an effective strategy, particularly if it is combined with each mix peering only with a small set of others. However this technique cannot provide an absolute protection against flooding since the attacker knows and controls the path through which a message is routed. Designs that disallow or restrict source routing could be a way forward to defending mix networks from flooding attacks.

3. DESIGN PRINCIPLES, ASSUMPTIONS AND CONSTRAINTS.

Using the analysis of Serjantov *et al.* one can calculate how much time, or how many messages, should be injected into a mix until an adversary can trace a message. While this can make an attack expensive, and will delay the overall functioning of the network, it does not guarantee that an attack will not succeed. On the other hand we will aim to completely eliminate the potential for $(n-1)$ or blending attacks.

Kesdogan *et al.* guarantee that most messages delayed will be dropped, but do not guarantee that single messages will not be traced. Again it would be easy to notice that such an attack is taking place (since messages are dropped) but no algorithmic way is included in the mix strategy that takes this into account. Therefore one of our aims will be to specify a way for the mix to detect that such an attack is taking place, and provide a strategy to counter it.

In designing RGB-Mixes to resist active attacks we will assume that the mixes have some knowledge of their environment, in particular the addresses, keys and capabilities of the other mixes in the network. This assumption is not unrealistic since clients require this information to send or receive anonymous messages, and directory server infrastructures are deployed to provide them [3]. We also require the RGB-Mix to be included in the list of active mixes in the directory listing and clients or other mixes to use it to relay traffic.

Furthermore we will assume that the network provides some anonymity, against the attacker. In most mix networks this means that the network is either not fully under the control of the adversary or that a large fraction of the mix nodes are honest. A key requirement is for the network to make indistinguishable to the attacker the *colour* of the traffic, which could be, as we will see, red, green or black.

While recognising that introducing dummy traffic into the network increases its cost, we do so for two purposes: first as signalling, to assess the state of connectivity with the rest of the network in the form of red traffic; and secondly in order to increase the anonymity sets while the mix is under attack, in the form of green traffic. It is a requirement that the amount of green traffic should be minimal (or even zero) if the mix is not under attack. On the other hand it increases when the mix is under attack in order to reduce latency, or to bootstrap the functioning of a network of RGB-Mixes.

4. RED-GREEN-BLACK MIXES.

An RGB-Mix receives a certain number of *black* messages per round or mixing interval. These are genuine messages to be anonymized or could be the product of a flooding attack mounted against the mix. The mix needs to estimate how many of these black messages are genuine in order to guarantee some quality of anonymity. Unfortunately because of the nested encryption, and the absence of identifying information in the mixed packets, the mix cannot do this by simple inspection.

In order to get an estimate of the number of genuine messages, a mix uses the same property that makes it unable to distinguish genuine from flooding traffic: namely that mixed traffic is not separable by a third party. With each output batch it includes a fraction of *red* messages, which are indistinguishable from other anonymous messages but are anonymously addressed back to itself. These messages are mixed with the outputs of the mix and are impossible to distinguish from other genuine black messages (notice that an attacker can distinguish them from flooding traffic). After a certain number of rounds or time intervals we expect the same fraction of red messages to come back to the mix. These messages can be distinguished by the mix since they were created by itself. This should be done in order to calculate their fraction in comparison with the black traffic received.

If the fraction of red messages received in a round or time interval is smaller than expected, subject to statistical fluctuations, this could mean one of two things. The mix could be under a blending attack, meaning that the genuine traffic is being blocked and only the attacker's messages are let through. Since the attacker cannot distinguish red messages from the genuine traffic it cannot selectively allow some of them through. Therefore it has to block them, and the fraction of red messages will drop depending on the severity of the attack. A second reason why the fraction of red messages could be small or zero is the fact that the mix has only recently started its operation and the red messages sent did not have enough time to loop back or traffic load is changing.

In case the fraction of red messages drops, a possible strategy would be to stop the operation of the mix until enough red messages are received, or forever if the attack persists. Unfortunately this transforms the blending attack to a denial of service attack on the mix. Furthermore if all the mixes implement this strategy it would be very difficult for the network to start its operation: all the nodes would block since their respective heartbeat red messages would not have yet arrived. This creates a deadlock situation.

Instead of employing the strategy described above dummy messages are introduced in order to guarantee the quality of the anonymity provided by the mix. A certain number of *green* messages are generated when necessary and injected in the output of the mix. These messages are multiple hop dummy messages that will be dismissed at the end of their journeys. Since an adversary is not able to distinguish them from other genuine black or red traffic these messages increase the anonymity set of the genuine messages trickled through by the attacker.

The objective we have set for the functioning of the mix is to reduce the amount of dummy traffic during normal operation, namely when the mix is not under flooding attack. The key to achieve this is to estimate the number of genuine black messages in the input, and only include green traffic if this is not above a threshold.

5. THE SECURITY OF RGB-MIXES.

The key to understanding the security of RGB-Mixes is the realisation that an attacker is not able to distinguish between red, green and black messages. Allowing through the red messages but not any genuine black message is difficult, and can only be done at random. On the other hand the RGB-Mix cannot distinguish the genuine black messages from the attacker's flooding messages, but can estimate their numbers using the calculated frequency of the red messages received during a mix round or time interval.

A number of messages $R+B$ is received by the mix during a period or round, with R being the number of red messages and B the number of black messages received. Out of the black messages some might be genuine traffic BT but some might be flooding traffic BF, with $B = BT + BF$. The probability of the adversary choosing a red message along with any genuine traffic chosen is equal to the fraction r of red messages included in the output of the mix. This assumes that the overall genuine traffic volumes do not change significantly.

An attacker will try to substitute genuine black traffic with flooding traffic that she can identify, thereby reducing the anonymity of the remaining message(s). If the substitution is done naively then no red messages will be received by the mix, which will use green cover traffic to maintain the sizes of the anonymity sets. Therefore an attacker will try to allow through some red messages. Since the attacker is colour blind she can only choose messages at random, according to the fraction injected in the network, until a certain number of red messages are present in the input batch.

The RGB-Mix needs to answer the following question: Given that R red messages are received, how many genuine traffic messages BT are likely the have been allowed through? The number of genuine messages that an attacker needs to chose for R red messages are present, if for each message the probability of being red is a fraction r, can be described by a negative binomial distribution.

$$\Pr[BT = x] = \binom{R+x-1}{R-1} r^{R-1}(1-r)^x \tag{1}$$

We can also calculate for a number R of red messages the expected number of genuine black messages, and its variance. Detailed derivarion of these can be found in [1].

$$E[BT] = \frac{R(1-r)}{r} \tag{2}$$

$$V[BT] = \frac{R(1-r)}{r^2} \tag{3}$$

The calculation above takes into account that the attacker is able to observe event where the mix receives a certain number of red messages. While an adversary is not able to tell that the message just input into the mix is red, she could still be able to observe some side effect, such as the mixing of a batch. This provides the mix designer with the flexibility to implement mixing strategies conditional upon the number of heartbeat messages received.

Let $(R + B)$ be the number of messages received in a batch and r the fraction of red messages sent by batch. Given that $(R+B)r$ red messages are expected during each round this would provide a standard anonymity set size of on average $\frac{((R+B)r(1-r)}{r} \approx B$. This number should be made large enough to provide adequate anonymity set sizes for the messages. In case the number of red messages received is smaller, then a number of green messages G' needs to be generated and output by the mix to make up for the potential loss of anonymity, where:

$$G' = \underbrace{\frac{((R+B)r)(1-r)}{r}}_{\substack{\text{Expected genuine black traffic}\\ \text{given total volume received}}} - \underbrace{\frac{(R)(1-r)}{r}}_{\substack{\text{Expected genuine black traffic}\\ \text{given number of red received}}} \tag{4}$$

$$= \underbrace{\frac{((R+B)r-R)(1-r)}{r}}_{\substack{\text{Difference is the number of green}\\ \text{dummies to be injected to}\\ \text{compensate}}} \tag{5}$$

$$R' = (R+B)r \tag{6}$$

Therefore if the mix is functioning properly and is not under flooding attack, it only outputs a minimal number of green, cover traffic, messages. When it is under attack it maintains the anonymity provided by outputting greater amounts of green cover traffic.

In case the attacker cannot observe the number of red messages in the stream reaching a threshold (such as a mixing batch being processed), a slightly different model can be used to estimate the number of genuine traffic messages BT. The probability a certain number of messages BT are

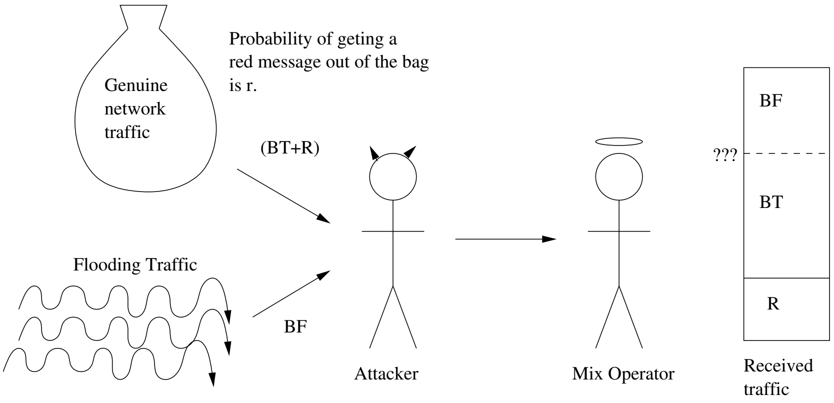

Figure 1: Model of the attacker and RGB-Mix

present in the batch, given that there is a number of red messages R and the probability a message addressed to the mix is red is r can be described as follows.

$$\Pr[BT = x|R, r] = \frac{\binom{x+R}{R} x^{1-r}}{\sum_{0 \leq x \leq B} \binom{x+R}{R} x^{1-r}} \tag{7}$$

A similar procedure to the first model can then be followed to estimate the deviation of the received genuine traffic from what would be expected if the number of red messages were indeed $(B + R)r$.

6. A CAUTIONARY NOTE.

The security of RGB-Mixes is calculated for the average case, namely the expectations are taken into account when calculating the amount of green traffic to be injected. This expected value will only be attained when the batch sizes are large enough, in comparison with the probability r a message received is red. Furthermore the analysis above is only accurate when the network traffic received by the attacker can be approximated by the red-black traffic bag, as shown in the figure. This means that the attacker taking a message from the network has a certain probability r of choosing a red message. In practise this is only an approximation since there is only a limited number of red messages, that will eventually run out if the experiment is repeated enough times. A more accurate model can be derived from the hypergeometric distribution.

Another critical assumption on which the models presented above are based is that the levels of genuine traffic do not change very much in time. Indeed there is no way a mix can tell the difference between an active attack and a genuine spike in traffic load. The traffic loads of the previous mixing rounds, or times, are therefore used to calculate the probability a red message is chosen by the attacker.

Another weak point of the method described above is that the attacker might try to influence r, the probability a message from the network is red. In order to avoid this the number of red messages injected in the network should be calculated based on a longer history of traffic load, than the information of the previous round of mixing. This way an attacker will have to attack for a very long time before getting any results.

The worst case presents itself when the mix does not receive any genuine traffic at all from the network the red messages are relayed. This means that the adversary will know which messages are red, and will be able to trivially perform flooding attacks, without being detected. The operational conditions under which this attack could be performed are a bit unusual. The mix under attack should not be included in the directory servers' lists, and therefore others should not be using it in order to relay traffic. Why would then the attacker try to attack it, since there is only minimal traffic on it? One reason could be that the attacker has lured a victim into sending a message through this particular mix. Again other methods of attack could be easier, such as forcing a victim to use a completely compromised node, instead of an "attackable" mix.

Finally it is worth noting that the green traffic offers some degree of protection against traffic analysis of the network, namely the traffic of a message node by node as it travels. It does not on the other hand offer any end to end protection against traffic confirmation. The green messages are simply discarded by mix nodes a few hops away, and modifying them to be sent to actual users is still a not very well understood problem.

7. FURTHER WORK AND FUTURE DIRECTIONS.

While sending red messages constantly might be interpreted as an inefficient usage of the mix network, it is worth noting that similar services are required, for different reasons. In the Mixmaster and Mixminion networks [6, 3] *pingers*, such as the one by Palfrader [7], are implemented in order to assess the state of the network nodes and links at all times. It would be interesting to study how the red traffic could be used to infer more information about the state of the network than used in the strategy described above. Collaboratively exporting such information and sharing it could also provide a distributed way of monitoring for attacks.

The study presented analyses a way in which mixes can assure themselves that they are connected to other mixes. This does not provide any assurance that any clients are actually connected to the network, or that any user traffic is input in the mix. It is an open problem to study how the model we presented can integrate users, and how this affects its overall complexity.

While some attacks could be difficult to detect for the node under attack, it might be the case that they are perceptible by third nodes. Again using the red traffic to detect other nodes in the network that might be under attacks and taking appropriate steps should be the subject of further study.

Finally the field of adaptive mixing strategies is rich in problems, since there is a lot of potential for an active adversary to influence the functioning of the mix. Strengthening such adaptive mix strategies while retaining their desirable characteristics, such as low latency and bandwidth efficiency, is still a challenge.

8. CONCLUSIONS

In this paper we have presented a dummy traffic policy that allows mixes to detect potential active flooding attacks and protect the anonymity they provide by generating dummy traffic. An analysis is presented on how a mix can infer the number of honest messages, based on a few network assumptions and the number of heartbeat messages it receives back. The field of active attack protection is quite challenging because of the strength of such attacks. On the other hand active attacks tend to be more detectable than passive ones, and we have shown that a mix can monitor for them and take steps to foil them after detection.

Acknowledgements The authors would like to thank Peter Palfrader for invaluable discussions about the intricacies of dummy traffic, Karen Veltri for assistance in proofreading, and the anonymous referees for corrections and suggestions.

9. REFERENCES

[1] Bhat, U. N. *Elements of applied stochastic processes.* John Wiley & Sons, Inc., 1972.

[2] Chaum, D. Untraceable electronic mail, return addresses, and digital pseudo-nyms. *Communications of the ACM 4*, 2 (February 1982). `http://www.eskimo.com/~weidai/mix-net.txt`.

[3] Danezis, G., Dingledine, R., and Mathewson, N. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy* (May 2003).

[4] Gulcu, C., and Tsudik, G. Mixing E-mail with Babel. In *Network and Distributed Security Symposium - NDSS '96* (1996), IEEE. `http://citeseer.nj.nec.com/2254.html`.

[5] Kesdogan, D., Egner, M., and Büschkes, T. Stop-and-go MIXes providing probabilistic anonymity in an open system. In *Information Hiding (IH 1998)* (1998), Springer-Verlag, LNCS 1525. `http://www.cl.cam.ac.uk/~fapp2/ihw98/ihw98-sgmix.pdf`.

[6] Möller, U., Cottrell, L., Palfrader, P., and Sassaman, L. Mixmaster Protocol — Version 2. Draft, July 2003. `http://www.abditum.com/mixmaster-spec.txt`.

[7] Palfrader, P. Echolot: a pinger for anonymous remailers. `http://www.palfrader.org/echolot/`.

[8] Serjantov, A., Dingledine, R., and Syverson, P. From a trickle to a flood: Active attacks on several mix types. In *Information Hiding (IH 2002)* (2002), F. Petitcolas, Ed., Springer-Verlag, LNCS 2578.