Threat Model

David Stainton

Table of Contents

Abstract

Here we describe the threat model of the katzenpost mix network transport protocol and also discuss threat models for addition protocol layers.

Overview of Mix Network Threat Model

Katzenpost is a decryption mixnet with no verification. In this context we can talk about individual mix nodes as a specialized kind of cryptographic router that has two properties other than routing messages to their destination:

1. bitwise unlinkability
2. latency

Bitwise unlinkability means that the adversary would not be able to link an output message with an input message based on the bits in the message. Likewise adding latency also helps prevent a network observer from correlating input messages with output messages on a mix node.

When mixes are composed in a network we get the Anytrust property which means that if a path (aka route) consists of at least one honest mix then there is still uncertainty for a passive network observer for the correlation between sent and received messages. There are many caveats here, see attacks below.

There is a hierarchy of security notions that are used by cryptographers to compare cryptographic primitives. Likewise, a hierarchy of privacy notions help us compare anonymous communication protocols. See NOTIONS for an in depth discussion and algebraic analsysis of privacy notions for anonymous communication protocols.

At the time of writing it is my understanding that the mixnet protocols we want to see in the world would have these privacy notions:

1. Sender Unobservability
2. Receiver Unobservability
3. Sender Receiver Unlinkability

The unobservability notions are a result of using decoy traffic whose use has limitations expressed in the anonymity trilemma.

Our threat model will consider the following eight categories of attacks:

1. n-1 attacks
2. epistemic attacks
3. compulsion attacks
4. tagging attacks
5. statistical disclosure attacks
6. denial of service attacks
7. timing attacks
8. cryptographic attacks (including considerations regarding cryptographic attacks by a sufficiently powerful quantum computational adversary)

As of the time of this writing, ALL mixnet attacks that we know about fit into one or more of the above categories or composite attacks, a combination of several of the above attacks.

1. statistical disclosure attacks

Adversary capability: In general for this particular attack category we assume that the adversary is a global passive adversary. However all of these attacks are actually possible with only a view of the perimeter of the mix network.

Adversary strategy: The adversary's goal is to capture as much of the social graph as they can by collecting sets of possible recipients of each sender and likewise sets of possible senders for each recipient.

Primary Mitigation tactic:

The best defense against intersection attacks is to use end to end decoy traffic which is sent and received by clients.

Imagine there is a mixnet with 10 Providers and only Alice and Bob are currently connected and sending messages. Alice does not send any decoy traffic. Mixes send loop decoys to themselves. Alice only sends messages to a Provider which Bob can retrieve messages from.

In this example we can say two things for certain:

1. mix decoy loops will be uniformly distributed over all 10 Providers
2. Alice's sent messages will go to 1 of 10 Providers

Therefore, without the use of end to end decoy traffic, a global adversary would be able to keep track of their observations of the network perimeter and learn statistical information about the social graph. In this example the uniform distribution of messages is perturbed by Alice's behavior on the network.

Mitigation tactic #1

In the context of Katzenpost/Loopix these classical style set intersection attacks don't work at full granularity because destination messages are queued on edge nodes (known as Providers) along with many other received messages to and from other users of the mix network.

Mitigation tactic #2

Many of our future protocols will scatter message segments across an ever changing set of Providers. This seems likely to increase uncertainty for adversaries.

Mitigation tactic #3

Clients will send decoy traffic such that the rate of all messages arriving on all of the Providers will be uniformly distributed even when the client is sending legit messages to a subset of Providers.

Mitigation tactic #4

Providers will modulate their decoy traffic send rate to be inversely proportional to the sum of all the rates of incoming messages from all clients directly connected to that Provider. In other words, when clients send zero messages the Provider sends a constant rate of decoy traffic. The Provider reduces it's decoy send rate when clients increase their send rate such that the total rate of messages coming out of the Provider remains the same if measured over a large enough period of time.

Conclusion

The sucess of a statistical disclosure attacks often has a lot to do with the advesary's ability to predict user behavior. Likewise if user behavior is very repetative and predictable then that might increase the probability that a statistical disclosure attack would work. These attacks could in theory take days/weeks or even months to perform depending on how much statistical information is leaked.

Statistical disclosure attacks such as short term timing correlation that the Tor network is known to be trivially vulnerable against do not in general apply to mix networks due to the added latency. However as latency is decreased we find ourselves pondering the Anonymity Trilemma ANONTRILEMMA which clearly states that Strong Anonymity is in opposition to low latency unless we send lots of decoy traffic. We need a formal methodogy for tuning the mixnet AND making the numerical calculations of the various tradeoffs that are the result of the mixnet tuning.

2. epistemic attacks

An epistemic attack refers to an attack where the adversary uses their knowledge of a mixnet client's knowledge of the network to their advantage. For example if Alice only learns of a subset of the network nodes then the adversary who knows this about Alice (or perhaps caused Alice to have partial knowledge) will be able to at least state some obvious conclusions such as: "messages sent along these routes are more likely to have come from Alice than any other client".

In general we mitigate this attack category by designing our key management and distribution (aka the dirauth system aka PKI) such that it shares the same information with all the clients.

3. compulsion attacks

Adversary capability: The adversary uses forceful means to procure the information they are after: violence, legal system, remotely compromising mix nodes using a zero day from the black market etc.

Conclusion

Reply Blocks (SURBs), forward and reply Sphinx packets SPHINX09 are all vulnerable to the compulsion threat, if they are captured by an adversary. The adversary can request iterative decryptions or keys from a series of honest mixes in order to perform a deanonymizing trace of the destination.

While a general solution to this class of attacks is beyond the scope of this document, applications that seek to mitigate or resist compulsion threats could implement the defenses proposed in COMPULS05 via a series of routing command extensions.

4. tagging attacks

There are many different types of tagging attacks. This is the only one I could think of that applies to Katzenpost, in an albeit contrived scenario.

Adversary capability

If the adversary is allowed to view the final payload decryption and can mutate the packet during it's transit then a 1 bit tagging attack is possible.

Adversary strategy:

Flipping a bit during transit would cause lots of bits to be flipped in each subsequent decryption set and thus the final payload integrity tag would be destroyed. So for the adversary, either the interity tag is intact or it is destroyed; this attack leaks 1 bit of information to the advesary.

Conclusion

This is Sphinx payload tagging attack is a result of the Sphinx design. However it's a very contrived example and we have trouble imagining it would apply in the real world.

Sphinx Payload Encryption Considerations

The payload encryption's use of a fragile (non-malleable) SPRP is deliberate and implementations SHOULD NOT substitute it with a primitive that does not provide such a property (such as a stream cipher based PRF). In particular there is a class of correlation attacks (tagging attacks) targeting anonymity systems that involve modification to the ciphertext that are mitigated if alterations to the ciphertext result in unpredictable corruption of the plaintext (avalanche effect).

Additionally, as the PAYLOAD_TAG_LENGTH based tag-then-encrypt payload integrity authentication mechanism is predicated on the use of a non-malleable SPRP, implementations that substitute a different primitive MUST authenticate the payload using a different mechanism.

Alternatively, extending the MAC contained in the Sphinx Packet Header to cover the Sphinx Packet Payload will both defend against tagging attacks and authenticate payload integrity. However, such an extension does not work with the SURB construct presented in this specification, unless the SURB is only used to transmit payload that is known to the creator of the SURB.

5. denial of service attacks

We don't have much defense against DOS attacks. Currently the Provider has a per client rate limiter that can be tuned by the dirauth system.

6. timing attacks

We probably have potential for many many timing attacks. Can we enumerate some of the more obvious and powerful timing attacks here?

7. n-1 Attacks

Adversary capability: Adversary is active and can send messages into the mix network AND the adversary can drop or delay messages sent to the mix network. Therefore the adversary has compromised the upstream routers for each of the perimeter mix nodes.

Adversary strategy: There are many variations of n-1 attacks and the one that works on Poisson mix strategy is this:

The adversary must delay or drop input messages to a given mix until they are reasonably certain the mix is empty before allowing the target message to enter and then exit the mix. The result of this attack is that the adversary learns where the target message is being sent.

Primary Mitigation tactic: Our theoretical defense is:

Each mix node uses a loop decoy heartbeat protocol to detect when an adversary is delaying or dropping input messages; that is, if the mix node doesn't receive it's own heartbeat loop message then it has detected this attack. A real world implementation would probably add some additional heuristics for example, the n-1 attack is detected when 3 heartbeats in a row were not received.

Our current status is:

- Mix loop decoy traffic is only implemented on interior mixes but it should also be implemented on Provders.
- The status of the decoy replies is ignored. Instead it should do bookkeeping and stop routing messages for some duration if certain heuristics are matched which include a threshold number of heartbeat messages not being recently received.

8. cryptographic attacks

This category should include not only merely breaking cryptographic primitives but also breaking the cryptographic protocols on a higher level of abstraction. One great example of this is the following attack on SURB usage described below.

SURB Usage Considerations for Volunteer Operated Mix Networks

Given a hypothetical scenario where Alice and Bob both wish to keep their location on the mix network hidden from the other, and Alice has somehow received a SURB from Bob, Alice MUST not utilize the SURB directly because in the volunteer operated mix network the first hop specified by the SURB could be operated by Bob for the purpose of deanonymizing Alice.

This problem could be solved via the incorporation of a "cross-over point" such as that described in MIXMINION, for example by having Alice delegating the transmission of a SURB Reply to a randomly selected crossover point in the mix network, so that if the first hop in the SURB's return path is a malicious mix, the only information gained is the identity of the cross-over point.

Appendix A. References

Appendix A.1 Normative References

ANONTRILEMMA

Das, D., Meiser, S., Mohammadi, E., Kate, A., "Anonymity Trilemma: Strong Anonymity, Low Bandwidth Overhead, Low Latency—Choose Two", IEEE Symposium on Security and Privacy, 2018, https://eprint.iacr.org/2017/954.pdf

COMPULS05

Danezis, G., Clulow, J., “Compulsion Resistant Anonymous Communications”, Proceedings of Information Hiding Workshop, June 2005, https://www.freehaven.net/anonbib/cache/ih05-danezisclulow.pdf

MIXMINION

Danezis, G., Dingledine, R., Mathewson, N., “Mixminion: Design of a Type III Anonymous Remailer Protocol”, https://www.mixminion.net/minion-design.pdf

NOTIONS

Christiane Kuhn, Martin Beck, Stefan Schiffner, Eduard Jorswieck and Thorsten Strufe, PETS 2019, https://petsymposium.org/2019/files/papers/issue2/popets-2019-0022.pdf

SPHINX09

Danezis, G., Goldberg, I., “Sphinx: A Compact and Provably Secure Mix Format”, DOI 10.1109/SP.2009.15, May 2009, https://cypherpunks.ca/~iang/pubs/Sphinx_Oakland09.pdf