Why is it so hard to send unobservable messages across the internet?

Martin Gwerder (m.gwerder@unibas.ch)
Department of Mathematics and Computer Science, University of Basel

Objectives

This work emphasizes on the requirements needed to create a system that is capable to transfer unobserved messages across the internet. It focusses on:

- What is required to create unobservable messages?
- What parts are already available as well established technologies?
- Where do we have a lack of reliable and researched technologies?

In this poster, I emphasize on the result. If you are interested at the argumentation, I recommend the corresponding paper referenced at the lower right for further reading.

Introduction

There are lots of works[1][2][3][4] that relate to anonymous message transfer. However – none of these works (with exception to TOR[1]) has been widely adopted in the internet. The reason for this is usually that peoples tend to concentrate on the method to transport the message and fail at the same time completely to take the real world and its problems into account. I collected some information that helps to create sensible and reliable systems.

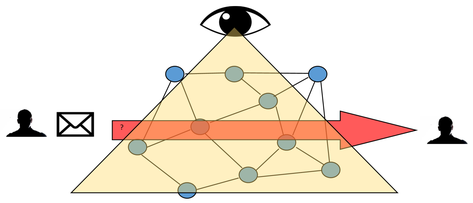

Figure 1: sending unobservable messages is not easy

Categorisation

In order to simplify the aspects of unobservability I categorized them into three categories.

- Acceptance
  What properties are expected from a user perspective? A "user" might be any participant (eg sender, recipient, or infrastructure administrator) except the suspected observer.
- Protocol
  What properties are expected when looking at the communication protocol? This point spans across all layers of protocol. It includes properties required for hiding communication, for routing and even for message interpretation.
- Infrastructure
  What properties are required from the infrastructure point of view? As infrastructure, we regard all publicly reachable parts of a message transfer system.

Acceptance

In order to get users to accept a solution we have minimum baselines to meet:

- Easy
  Users might reject a solution that is hard to learn. Experiences with new systems show that when covering a field that is already covered by a convenient system, the willingness of users to learn for the benefit of security related issues is very low.
- Fast
  Email, chat and similar systems are available fast and convenient. Users are not willing to accept a system where a message transfer might take hours or days.
- Reliable
  Available message transferring systems are quite reliable. This makes it hard for new systems. They have to be equally reliable in order to be accepted.
- Not abuseable
  If a new system is misused easily (eg. for spamming purposes) the acceptance drops drastically. This is because misuse of today's mass message transfer systems already leads to considerable annoyance.

Protocol

In order to succeed with the goal the protocol needs to support certain features:

- Unidentifiable
  If a message or a participating system is automatically identifiable then it is easy for an adversary to block some or all parts of the infrastructure. Only a service that is able to hide its messages in legitimate network traffic is not subject to selective blocking.
- Untagable
  If messages going thru the system are taggable by any of the routing participants then anyone might tag messages and then follow them thru the network.
- Unreplayable
  If an adversary can replay any part of the message, he can identify the traffic generated by those messages by statistical means.
- Monolythic messages
  If a message is not self-contained then "bugging" is an easy way to identify the message on its way up until they reach the recipient.

Infrastructure

In order to succeed there are as well certain baselines for the infrastructure:

- Unknown endpoints
  Every endpoint should behave the same as an intermediate routing point. They should receive and send messages so that they are not identifiable as endpoints. Identifiable endpoints simplify analysis.
- No relations between single hops
  Messages transferred from server to server must be unrelated. Server identifiable to send messages due to received messages are potential targets for analysis.
- Untrusted infrastructure
  Unlike in a company owned net, in the internet trusting an infrastructure is not possible. It is very often not clear who owns a server and who else does have access to it. So an unobservable system may not build its unobservability based on behavior of the transporting infrastructure.
- No central infrastructure
  Central infrastructure may be attacked or shut down. They are easier to monitor than an unknown number of participants.
- No direct communication between endpoints
  If sender and receiver communicate directly then they are easily identified.

Important factors

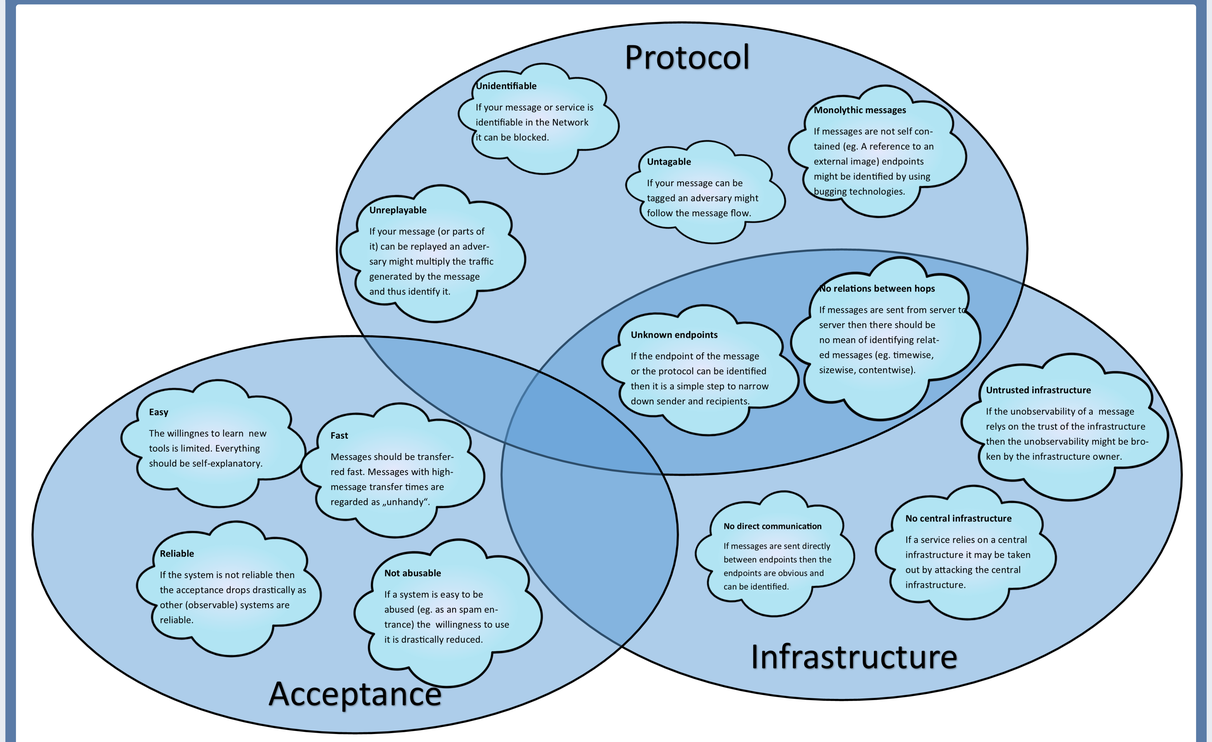

Figure 2: Important factors when designing an unobservable message channel

Conclusions

Sending unobservable messages thru a public network is not easy. It cannot be done by inventing a new identifiable service. It has to blend into today's traffic and look unsuspicious compared to all other traffic to be of any value. Yet it has to be easy to handle and simple to understand. Combining today's technologies might be sufficient but have to be researched further.

Additional Information

For additional information, please see the corresponding papers and presentations published at: https://www.gwerder.net/~mgwerder/phd/how_unobservable/ or use the QR code in the top right corner.

References

[1] Roger Dingledine, Nick Mathewson, and Paul Syverson.
Tor: The second-generation onion router.
In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[2] Ulf Möller, Lance Cottrell, Peter Palfrader, and Len Sassaman.
Mixmaster Protocol — Version 2.
IETF Internet Draft, July 2003.

[3] Shlomi Dolev and Rafail Ostrobsky.
Xor-trees for efficient anonymous multicast and reception.
ACM Trans. Inf. Syst. Secur, 3(2):63–84, 2000.

[4] Brian Neil Levine and Clay Shields.
Hordes — a multicast based protocol for anonymity.
Journal of Computer Security, 10(3):213–240, 2002.