

Closed-Circuit Unobservable Voice Over IP.

Carlos Aguilar Melchor
XLIM-DMI
Université de Limoges
123, av. Albert Thomas
87060 Limoges Cedex
FRANCE

Yves Deswarte
LAAS-CNRS
Université de Toulouse
7, avenue du Colonel Roche
31077 Toulouse Cedex 4
FRANCE

Julien Iguchi-Cartigny
XLIM-DMI
Université de Limoges
123, av. Albert Thomas
87060 Limoges Cedex
FRANCE

Abstract

Among all the security issues in Voice over IP (VoIP) communications, one of the most difficult to achieve is traffic analysis resistance. Indeed, classical approaches provide a reasonable degree of security but induce large round-trip times that are incompatible with VoIP.

In this paper, we describe some of the privacy and security issues derived from traffic analysis in VoIP. We also give an overview of how to provide low-latency VoIP communication with strong resistance to traffic analysis. Finally, we present a server which can provide such resistance to hundreds of users even if the server is compromised.

Index Terms

Unobservability, Anonymity, Voice over IP, Low-Latency

1 Introduction

In an IP network, a communication is composed of packets. Each packet has a set of headers and a content. When confidentiality is a concern, in particular with respect to eavesdropping, it is usually assumed that the eavesdropper is interested in the contents of the packets. However, sometimes, an eavesdropper will be just interested in the packet headers (to learn who are the sender or the recipient for example) or in their presence (to detect an ongoing communication or changes in the amount of traffic on the network).

The existence of a communication, when it does begin or end, the users taking part in it, or the amount of information exchanged, is part of the *meta-data* defining a communication. Hiding the contents of a communication is easy to achieve through the encryption of the content of each packet. Hiding the meta-data, on the other side, can

be very difficult. End-to-end encryption cannot be used on the packet headers as they are needed by the intermediate nodes of the network for routing purposes. Moreover, in most of the networks over which IP is implemented an attacker eavesdropping on a communication link will be able to observe the existence of all the transiting packets. The systems that try to hide the meta-data associated to a communication are called *anonymous communication systems* and the act of trying to discover this meta-data is called *traffic analysis*.

Strong traffic analysis resistance is hard to obtain. It is commonly accepted that the only practical way to achieve it is by the usage of relays that hide the meta-data of the communications.¹ There exists an extensive literature on how to use multiple relays sequentially to obtain traffic analysis resistance. Some of the proposals are based on an usual server-client model [15, 10], and others are peer-to-peer [8, 17], but only two finalized implementations are currently widespread and operational: JAP [7] the Java Anon Proxy, and Tor [6], the second generation onion routing network.

In order to respect the latency constraints of VoIP communication it is not possible (at least over the Internet) to use multiple relays sequentially, and therefore it is preferable that all the users communicate with only one relay. In this paper, this relay is called the communication server. We consider that packets can be routed through this server with a reasonable round-trip time for VoIP communication (including at most 100 ms of processing time in the server). We have not tested or implemented the servers we propose in this paper. We aim to present a theoretical overview of what performance we can achieve with different techniques.

We have also decided to focus on the communication stream. We do not consider signaling issues. How to define a practical signaling protocol that avoids traffic analysis is well beyond the scope of this paper, as well as how

¹In 1985 a relay-independent approach was proposed [14, 16]. However the anonymous communication systems derived from this technique with practical implementations have been based on relay usage [12].

