TorCloak Infrastructure Draft

Francisco Silva, Diogo Barradas, Nuno Santos

August 2022

1 Introduction

TorCloak consists of a new Tor pluggable transport (and surrounding ecosystem) that will establish covert channels between Tor clients and TorCloak bridges, which act as proxies to the free Tor network. TorCloak will set up these covert channels by concealing Tor traffic on the video streams of widely-used web conferencing services based on WebRTC technology, e.g., Jitsi Meet. TorCloak uses Protozoa [1] as the base mechanism to perform this covert data transmission. The basis of the mechanism is to intercept the outgoing video frame bytes and replace them for streams of Tor packet bytes generated by the Tor Browser. Thus, under the guise of a regular video call, users will be able to freely and stealthily access services like SecureDrop without being blocked or detected. Covert channels created this way will prevent a censor from distinguishing TorCloak traffic from unmodified WebRTC streams using deep-packet-inspection (DPI) or statistical traffic analysis, a cornerstone for the successful deployment of previous pluggable transports.

For making this approach both easy to use and to scale, we require an efficient mechanism to connect our TorCloak clients and bridges. This mechanism must enable Tor users to find and connect to TorCloak bridges using our custom protocol. It is also important that this mechanism makes it hard for a potential censor to find such bridges and block them. The rest of the document explains our current proposed solution.

2 Proposed Solution

2.1 Required data to disseminate TorCloak bridges

In TorCloak, both the client and the bridge must join a given videoconferencing room. For this to happen, two main values must be exchanged:

- Chatroom IDs, e.g., **https://meet.jit.si/torcloak1234**: These are URLs of the host application (in this example, Jitsi Meet) and a videoconferencing room id (e.g., **torcloak1234**), that can be defined by the TorCloak bridge operator.

- the videoconferencing room password (so that only the bridge and the client can access it)

2.2 Establishment of TorCloak sessions

Figure 1 shows a TorCloak Client establishing a covert session with a TorCloak bridge and using it to access `cnn.com` through the Tor Network. This involves the following steps:

(A) When the bridge is set up it registers itself with the BridgeDB, indicating a) the chatroom id (**meet.jit.si/torcloak1234**) and b) the chatroom's password (**password1234**).

(B) The bridge then joins the chatroom and waits for a client to connect.

(C) Upon start up, a Tor Browser configured to use the TorCloak pluggable transport requests a TorCloak bridge's chatroom ID and password from the BridgeDB.

(D) After receiving this data, the TorCloak client can now join the chatroom together with the TorCloak bridge and the covert media session can be established.

(E) Tor browsing data can now be covertly exchanged through the media session, as explained in Section 1.

(F) After this, the user can freely access websites, as usual (in this example `cnn.com`). TorCloak Client will encode the requests into the media session and deliver them to the bridge. The TorCloak Bridge will then decode the request and forward it to the Tor Network.

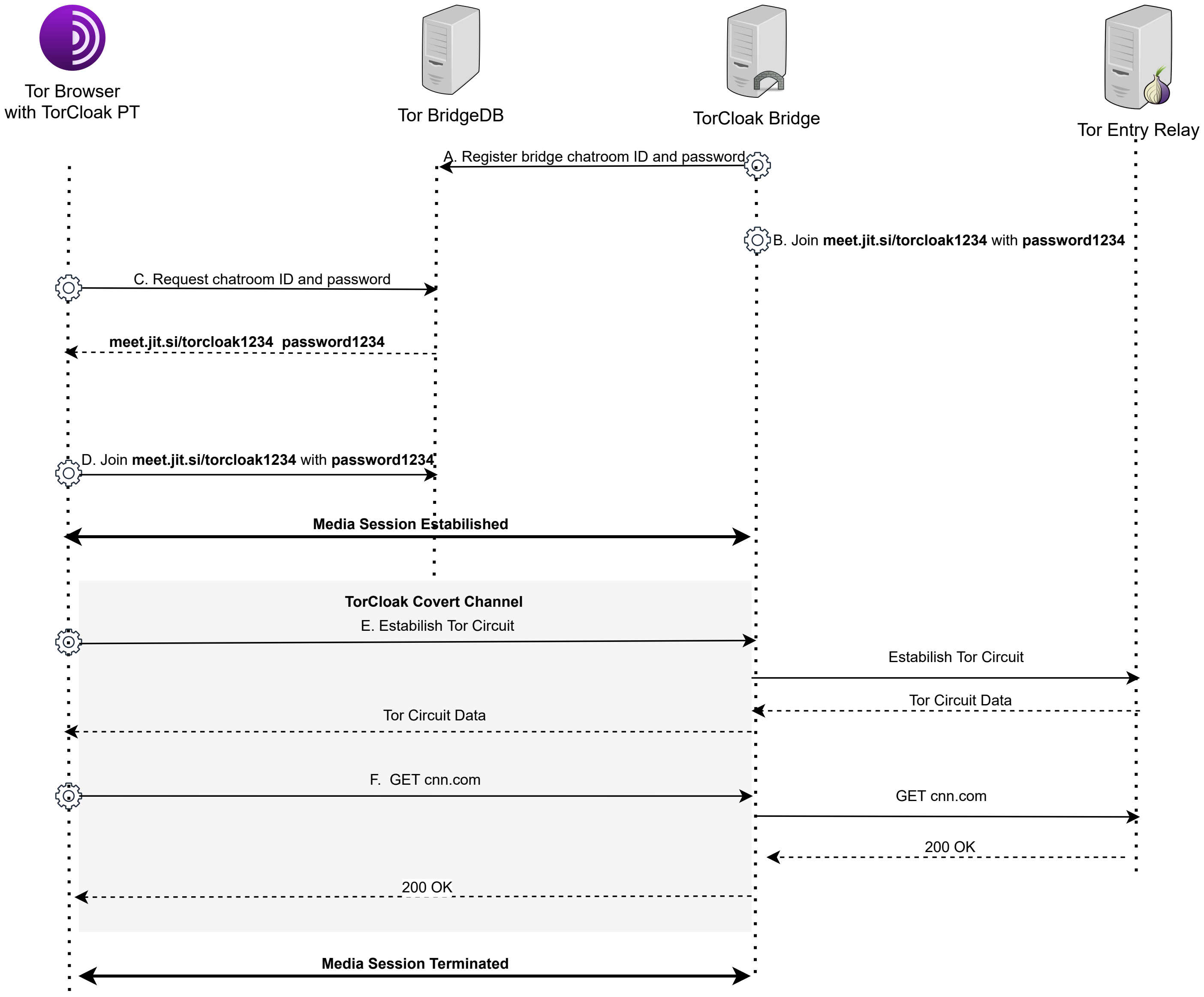
(G) When the user is done, the session can be terminated and the chatroom is tear down. Optionally the chatroom ID can be rotated after it has been used. This can be configured to diminish the changes of a censor being able to enumerate the chatroom IDs used by a TorCloak client or bridge.

This solution is geared towards leveraging the bridge distribution infrastructure already made available by the Tor project and make the process of establishing a bridge connection as simple as possible for the user, automating most of it. Put briefly, instead of exchanging a Tor bridge's IP address like in existing Pluggable Transports, e.g., obfsproxy, we wish to share videoconferencing meeting points for our video call sessions that covertly exchange Tor traffic.

By making use of existing bridge distribution mechanisms, it is our wish to increase our tool's compatibility and reach the most amount of users possible. TorCloak's use of Tor's BridgeDB infrastructure would organically benefit from the performance and security benefits already provided by BridgeDB.

References

[1] D. Barradas, N. Santos, L. Rodrigues, and V. Nunes. Poking a Hole in the Wall: Efficient Censorship-Resistant Internet Communications by Parasitizing on WebRTC. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2020.



Tor Browser
with TorCloak PT
Tor BridgeDB
TorCloak Bridge
Tor Entry Relay
A. Register bridge chatroom ID and password
B. Join **meet.jit.si/torcloak1234** with **password1234**
C. Request chatroom ID and password
meet.jit.si/torcloak1234 password1234
D. Join **meet.jit.si/torcloak1234** with **password1234**
Media Session Estabilished
TorCloak Covert Channel
E. Estabilish Tor Circuit
Estabilish Tor Circuit
Tor Circuit Data
Tor Circuit Data
F. GET cnn.com
GET cnn.com
200 OK
200 OK
Media Session Terminated

Figure 1: Covert session