Privacy preserving Distributed Hash Tables

Gonçalo Pestana (goncalo@hashmatter.com)

DRAFT

DRAFT: This document is work in progress. Please send your comments, suggestions and corrections to gpestana@hashmatter.com

In this paper we consider privacy threats of Distributed Hash Tables (DHT), more specifically what data is leaked when requesting content in DHT and how. In addition, we consider mechanisms that could prevent private information leakage in DHTs and discuss how current projects would benefit from a privacy preserving DHT.

Introduction

Distributed Hash Tables are basic building blocks for the decentralized web and P2P systems. It allows multiple peers to collaboratively function like a hash table, where each peer can store and lookup for `key:value` pairs in a P2P, decentralized network.. Its decentralized nature provides scalability and resilience, while, on the other hand, leaks information about what `key:value` pairs peers are requesting from the network. As P2P and decentralized networks gain popularity, it is important to design and implement networks which not only preserves its users' privacy, but also do deliver scalability and performance. Failing to deliver on those properties will render decentralized networks unusable and unattractive for mass adoption and applications that can impact positively the current technology landscape.

By design, the same mechanism that made DHT possible to work effectively as a scalable and well performing P2P network is also responsible for leaking private information about network peers.

Despite the existence of different flavors of DHTs such as [ref, ref, ..], it is possible to abstract the DHT lookup mechanism as follows: Consider a lookup initiator peer `P_i` who wants to access a key `KEY` stored in the network by a peer `P_store`, which ID is the closet to `KEY` in the network's key domain. The lookup initiator maintains a table with a set of peers - its finger table - which is its current view of the network. First, `P_i` send a request to its neighbor peers to return the set of nodes in their finger-table that are closer to `KEY`. Once `P_i` gets this information, it selects the set of nodes closer to `KEY` ID and repeats the process. The recursive lookup algorithm makes it possible for `P_i` to use other peers' finger-tables to resolve the query to `KEY`.

Although the lookup mechanism in DHTs enables a scalable and churn-resilient P2P network of cooperative peers, it also leaks information about what data a lookup initiator is looking for in the network. It is relatively cheap for passive adversaries to gain information about what a lookup initiator is searching in the network in two levels: 1) by inspecting intersected lookup request packets and 2) by following the lookup path of request packets, since it will converge to the peer whose ID is closest to the lookup key.

In addition to leaking information through the lookup mechanism, once the lookup initiator resolved `P_store`, `P_i` must directly request the `KEY` from `P_store`. Unless there is a mechanism that hides `P_i` ID while, at the same time, making it possible for `P_store` to deliver the requested content, `P_store` will learn that the lookup initiator is interested in `KEY`.

In this paper, we focus on the privacy threats of passive attackers that gather and correlate lookup requests in order to link lookup requests with its peer

initiator and key. We also consider honest peers which may also link peer initiator and key information due to how DHT lookup mechanism works

Threat model

We focus on the privacy threat of DHT traffic analysis by passive adversaries and nodes which participate honestly in the network.

An adversary controls a set of well-behaving nodes which are part of arbitrary but well-distributed network segments. Each node collects all routing and lookup requests from the network, which can be correlated. The adversary can correlate all collected data to link peer IDs to requested key IDs. The data classes that allow an adversary to link peer IDs to requested key IDs are:

- **lookup request packets**: The lookup request packets contain information about which key a peer ID is looking for.
- **lookup request paths**: By looking at the flow of lookup requests, the adversary can understand which segment of the network the requests are converging to, ultimately leaking information about which key was requested.

By design, an honest peer which stores the pair `key:value` requested will have access to which peer initiated the request. Although this information is necessary for replying to `GET` requests, we consider this a privacy threat.

Goals

- An adversary who controls a fraction of the network nodes is not able to link lookup requests to the lookup initiator by passively listening to lookup requests across network segments and correlate them;
- Peers providing content stored in the network are not able to de-anonymize the peer who requested the data.

Current potential applications

- IPFS
- Hyperswarm
- DHT support on ZeroNet
- Lightening network
- DHT-based relay directory for Tor
- Hidden service directory for Tor

Privacy preserving mechanisms for DHTs

In this section we outline ideas of mechanisms that would improve privacy in DHT based on the threat model and goals described in the sections above.

- In-DHT onion routing:
- Friend-to-Friend routing::

Literature review

Open questions and future work

Conclusions