Privacy preserving lookups with In-DHT Onion Routing [RFC]

DRAFT: This document is work in progress. Please send your comments, suggestions and corrections to gpestana@hashmatter.com or join the conversation at ((gpestana), n.d.)

Abstract: In this paper, we consider design and implementation details for using onion routing to preserve privacy of lookup initiators in Distributed Hash Tables (DHTs). We also review literature and outline open challenges and future work to achieve privacy preserving DHT which is secure, scalable and decentralized.

Introduction

Distributed hash tables

Distributed Hash Table (DHT) is a network overlay that implements a hash table over a P2P network. The user API with has two primitives: `store(data)` and `lookup(ID)`, which allows peers to store and lookup for data in the network. While each node in the network is assigned with an unique ID belonging to a certain ID domain, the data stored in the network is also uniquely identified with an ID belonging to the same ID domain. This property is the basis to resolving the location of content in the network, since the peers with the closest ID of a certain data chunk are responsible for storing it. The gist of any DHT is its routing protocol, which defines how peers collaboratively pass requests within each other in order to resolve the correct peer where data is stored (or where to store the data, depending on what action is taking place). Although different flavors of DHT protocols implement routing in different ways, the many

This design enables peers to coordinate itself to store and retrieve keyed values without central point of authority or registry and provide a decentralized.

The problem: lookup initiator privacy

When peers make a network request for a specific data ID, usually a chain of network requests starts in order for the lookup initiator to learn which peer in the network has the data - or if the data is not available. While the open collaborative nature of DHTs enables large networks of peers to resolve requests without a a single point of failure and central authority, it also leaks information to anyone in the network regarding what peers are requesting. Potentially any peer in the network has access to which data is requested by any network peer. On applications where sensitive data is stored and requested, this poses a privacy problem to every peer in the network and renders the whole network unusable from a privacy perspective.

A (potential) solution: In-DHT onion routing

Onion routing allows bidirectionally traffic with reduced latency in P2P networks and gets its protection from creating cryptographic circuits along routes that an adversary is unlikely to observe and/or control. These properties make onion routing a good fit for DHTs routing where the lookup initiator is private.

In-DHT onion routing consist of encrypting the lookup request in multiple layers and relaying it through a path of peers that are able to decrypt each layer and forward the remaining packet to the next hop in the path.

The first step for a peer to use in-DHT onion routing is to construct an onion circuit, An onion circuit is the set of peers that will be as relays and will be decrypting and forwarding the onion packet in the network. Tor uses a central authority directory with information necessary to construct the onion circuit.

We aim at relying entirely on a P2P, decentralized network to securely build the onion circuits.

Threat model

Onion routing security breaks against an adversary that can observe the whole network and . Onion routing is designed to be secure against adversaries with only partial view of the network, so we do not consider an attacker that have a full view of the network at any time and can see all links between peers (or, as (Syverson 2013) refers to it, "The Man").

We consider a local adversary with partial view of the network and that controls up to a fraction `f` of colliding nodes in the network. We consider `f` to be `0.2`.

Goals

In the current work, we define as main goal to design a DHT which leverages onion routing to achieve the following goals:

- Provide anonymity and message unlinkability to lookup initiators;
- Secure against local adversary with partial view of the network that can control up to 20% of the network (see threat model above);
- Network information management infrastructure (e.g. peer relay directories, PKI infrastructure, etc) are completely decentralized;
- Low latency network;
- Relatively low computational and time overhead;

In-DHT onion routing

Protocol

Onion routing vulnerabilities

Previous research have demonstrated that onion routing is vulnerable to a set of de-anonymizing attacks under the assumption that an active and passive attacker with partial view of the network - and in some cases even weaker. These attacks are effective given the low-latency nature of the protocol. The In-DHT onion routing is vulnerable to the same set of attacks, which are:

- **Adversary controls entry and exit relay**: this timing attack is effective regardless of the entropy in the network (Syverson 2013). An attacker which controls entry and exit relay is able to correlate and de-anonymize packets in the circuit, regardless of the number of peers using the same relays.

- …

Note that low latency anonymity networks - such are onion routing-based netowkrs - are fundamentally broken against "The Man" (Syverson 2013). They do offer, though, some protection against weaker adversaries and may be an interesting trade-off between latency, overhead and anonymity for the DHT use case.

Onion circuit building

An attacker can deanonymize an onion routing user if 1) the circuit is known to the attacker or 2) the attacker controls both entry and exist relays in the path. Thus, the onion circuit building is an important part of the onion routing security. In this section we define circuit building security in the context of in-DHT onion routing, review literature on the subject and outline open challenges and future research work.

Provably secure onion circuit building

Conceptually, a lookup initiator must be able to hide from internal and external peers which nodes were selected when constructing the onion circuit. Formally that is translated by saying that the probability for an adversary node to successfully guess the IDs of the set of nodes which form a circuit to be uniformly distributed in the network, that is:

- given a set `N = {n, n_1, ..., n_i-1}` of `i` nodes that exist in the network in a given time;

- a set of nodes `R = {n, n_1, ..., n_l-1}` which form an onion circuit with a set of nodes of length `l`,
- the length of the circuit `l`;
- and `Pr(n)`, which represents the probability that an adversary can successfully guess that a node `r` is a relay chosen to be part of a circuit, must be:

  `Pr(n) = l/sum(N)`

The probability must hold even though:

1) the node `n` does cannot have a full picture of the network at a given time and;
2) the adversary node knows the overlay fingertable of `n` at any time and;
3) relay nodes enter and leave the network in unpredictable ways (churn).

This means that

Onion circuit construction vulnerabilities

Selecting the nodes that will be part of the onion circuit in a completely decentralized manner is no easy tasks. (to explain: relays are not kept anywhere and should be queried as a peer joins the network or wants to build an onion circuit)

Secure

Approaches against route capturing and bias attacks when getting information to construct the onion circuit are:

1) Identify and isolate nodes performing attacks; This is not trivial to achieve in a completely decentralized way (i.e. without relying on central certificate authorities);

2) Use mechanisms to find node IDs that do not rely on direct finger table sharing from other nodes. One naive example could be to passively listen to networks activity for a while and select the node IDs that are more likely to be honest. This approach has the problem of publicly narrowing down too much the subset of nodes that are eligible to be part of the circuit, since possible malicious nodes can learn which nodes are more likely to be picked by the circuit constructor when using this passive mechanism.

Open questions and future work

minimum required entropy:

secure and decentralized circuit construction:

incentives:

public key distribution:

sybil attacks and in-dht routing:

peer discovery and circuit building:

performance and overhead:

Previous work

(Johnson and Syverson 2009) presents how trust information can improve the anonymity provided by onion-routing networks. The work presents a trust model used for selecting secure relays that reduce probability that adversaries can control both entry and exit nodes.

Bibliography

(gpestana), Goncalo Pestana. n.d. "Metadata Resistant Dht - Github." *Metadata Resistant DHT - Github*. https://github.com/gpestana/notes/issues/8.

Johnson, A., and P. Syverson. 2009. "More Anonymous Onion Routing Through Trust." In *2009 22nd Ieee Computer Security Foundations Symposium*, 3–12. https://doi.org/10.1109/CSF.2009.27.

Syverson, Paul. 2013. "Why I'm Not an Entropist." In *Security Protocols Xvii*, edited by Bruce Christianson, James A. Malcolm, Vashek Matyáš, and Michael Roe, 213–30. Berlin, Heidelberg: Springer Berlin Heidelberg.