Onion Routing

by

Harikrishnan S (M.Tech CSE)
Ramji Nagariya (M.S CSE),
Sai Sambhu J (M.Tech CSE).

1) Introduction

Onion routing is an infrastructure for private communication over a public network. Traffic analysis can be done to reveal who is talking to whom. Anonymous connections provided by Onion routing are designed to be resistant to traffic analysis.

In onion routing, instead of making socket connections directly to a responding machine, initiating applications make connections through a sequence of machines called onion routers. The onion routing network allows the connection between the initiator and responder to remain anonymous.

For any anonymous connection the sequence of onion routers in a route is strictly defined at connection setup. But each router can identify only the previous and next hops along a route.

2) Operations

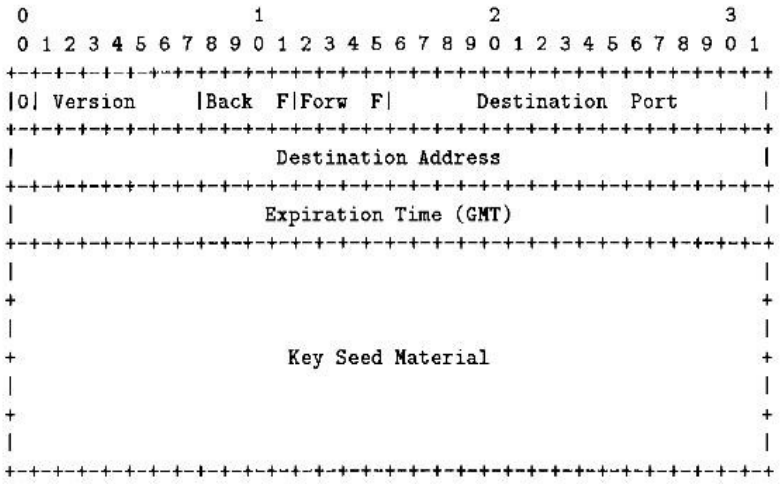
The onion routing network is accessed via a series of proxies. An initiating application makes a socket connection to an application proxy. This proxy massages connection message format (and later data) to a generic form that can be passed through the onion routing network. It then connects to an onion proxy, which defines a route through the onion routing network by constructing a layered data structure called an onion. The onion is passed to the entry funnel, that occupies one of the longstanding connections to an onion router and multiplexes connections to the onion routing network at that onion router. That onion router will be the one for whom the outermost layer of the onion is intended. Each layer of the onion defines the next hop in a route. An onion router that receives an onion peels off its layer, identifies the next hop, and sends the embedded onion to that onion router. The last onion router forward data to an exit funnel, whose job is to pass data between the onion routing network and the responder.

The direction in which the onion travels is the *forward direction* and the reverse is *backward direction*.

3) Using Onion Routing

3.1 Connection Setup

To build the anonymous connection to the exit funnel, the onion proxy creates an onion. An onion is a multi-layered data structure that encapsulates the route of the anonymous connection starting from the onion router for that exit funnel and working backward to the onion router at the entry funnel. Each layer has the structure shown below.



Drawing 1: Single Onion Layer

During connection setup, at each router, after peeling of the outermost layer, that router gets a key seed also along with the next hop information. This key seed is used to make two keys, one to be used for encryption in *forward path* during data transfer and the other one for encryption in the opposite direction.

3.2 Data Movement

Once the anonymous connection is established, it can carry data. Before sending data over an anonymous connection, the onion proxy adds layers of encryption for each onion router in the route by applying the inverse of the forward cryptographic operations. As data move through the anonymous connection, each onion router removes one layer of encryption,so it arrives at the responder as plain text. This layering occurs in the reverse order for data moving back to the initiator. Therefore data that have passed backward through the anonymous connection must be repeatedly post crypted to obtain plain text.

The onion proxy must repeatedly crypt data to either add the appropriate layers of cryption on outgoing data, or remove layers of cryption from incoming data.

3.3 Destroying the connection

Just as socket connections are torn down, anonymous connections need to be destroyed when the connection is broken. An onion router that decides to tear down a connection sends a destroy message forward and backward along the anonymous connection.

3.4 Reply Onions

If an initiator expects a later reply from the responder? An obvious solution is to keep the anonymous connection open. This may not always be practical. Another solution is a reply onion. An initiator's onion routing proxy can create a reply onion that defines a route back to him.

A reply onion can be used by a responder to create an anonymous connection back to the initiator at a later point in time.

3.5 Reply Onion Demonstration:

(Initiator) ->W ----- X ----- Y ----- Z ----> (Responder)

(Initiator) --W <----- X <----- Y <----- Z <---- (Responder) – Reply onion path

W, X, Y, Z are onion routers.

Responder has the reply onion from passed on from initiator.
Responder sends reply onion to onion routing proxy Z.

- Z peel off a layer. Gets the key seed intended for itself and an onion to be passed to Y.

Z sends the onion to Y.

- Y peels of the next outermost layer and gets the keyword for itself and an onion.

This process repeats back until it reaches initiator's onion proxy.
As this reply onion travels back to the initiator' proxy, an anonymous connection is getting constructed incrementally.
Once the reply onion reaches the initiator's proxy, the proxy extracts all the keys in it. These keys correspond to keys given to each onion router on the path from responder back to initiator's onion proxy.

Now data movement can happen as usual, as keys are available at the appropriate locations.

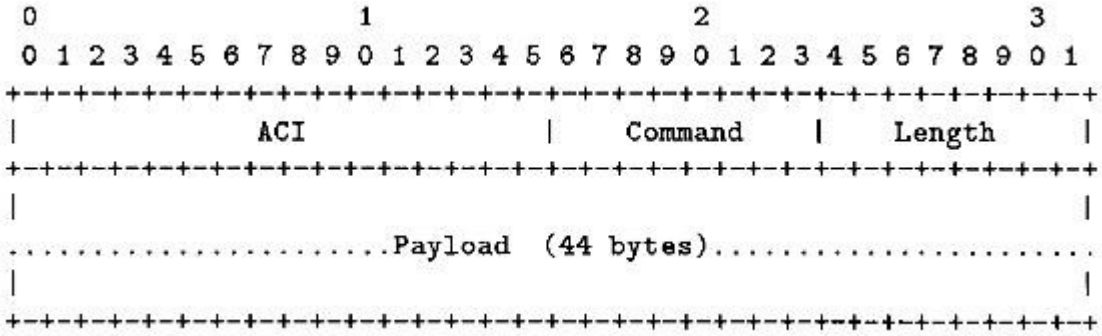
4) Network

Although we call this system onion routing, the routing that occurs here does so at the application layer of the protocol stack and not at the IP layer. More specifically, we rely upon IP routing to route data passed through the longstanding socket connections. An anonymous connection is comprised of portions of several linked longstanding multiplexed socket connections. Therefore, although the series of onion routers in an anonymous connection is fixed for the lifetime of that anonymous connection, the route that data actually travels between individual onion routers is determined by the underlying IP network. Thus, onion routing may be compared to loose source routing.

5) Onion Router Interconnection

During onion network setup (not to be confused with anonymous connection setup), longstanding connections between neighboring onion routers are established and keyed. The network topology is predefined and each onion router knows its neighbors.

Communication between onion routers is packaged into fixed sized cells, that allows for the multiplexing of both anonymous connections and control information over the longstanding connections.



Drawing 2: Inter Router Communication Cell Structure

The Anonymous Connection Identifier (ACI) and Command fields are always encrypted using the link encryption between neighboring nodes.

There are four type of cells in onion routing system.

- PADDING (0)
- CREATE (1)
- DATA (2)
- DESTROY (3)

Each anonymous connection is assigned an ACI at each onion router, which labels an anonymous connection when it is multiplexed over the longstanding connection to the next onion router. ACI's must be unique on their longstanding connection but need not be globally unique.

Data moves through an anonymous connection in DATA cells.

If a connection is broken, a DESTROY command is sent to clean up state information. The ACI field of the DESTROY command carries the ACI of the broken connection. The length and payload must be random. Upon receipt of a DESTROY command, it is the responsibility of an onion router to forward the DESTROY appropriately and to acknowledge receipt by sending another DESTROY command back to the previous sender.

The PADDING command is used to inject data into a longstanding socket to further confuse traffic analysis. PADDING cells are discarded upon receipt.

6) Vulnerabilities

Onion routing is not invulnerable to traffic analysis attacks. With enough data, it is still possible to analyze usage patterns and make educated guesses about the routing of messages. Passing dummy traffic may help to an extent in this case.
If an attacker observes a relatively under-loaded onion router, he or she can link incoming/outgoing messages by observing how close together in time they are received and re-sent. Thus a timing analysis is possible.
Intersection attacks rely on the fact that onion routers periodically fail or leave the network; thus, any communication path that remains functioning cannot have been routed through those routers that left, neither can it involve routers that joined the network recently.

In a predecessor attack, an attacker who controls an onion router keeps track of a session as it occurs over multiple path reformations (paths are periodically torn down and rebuilt). If an attacker observes the same session over enough reformations, he or she will tend to see the first router in the chain more frequently than any other router.

Onion routing exit nodes give the operator complete access to the content being transmitted and therefore the onion network should not be used to transmit sensitive information without using end-to-end cryptography.