Looking For Group: Open Research Questions about I2P

Jack Grigg
I2P
str4d@i2pmail.org

1. INTRODUCTION

There are two active implementations of general-purpose onion routing-based privacy networks: Tor [3] and I2P [1]. There are certain similarities between the two networks (onion routing being one), but also various differences in their overall approaches, arising in part from their differing origins. While onion routing has been the subject of academic research for decades, the vast majority of research focuses on Tor's approach.

In this talk, I will give a brief overview of I2P, and then present several key areas where its design choices differ from Tor. By examining and contrasting the assumptions these networks are based on, I hope to illuminate as-yet unexplored research avenues, that can benefit both I2P's userbase and the wider community.

2. I2P OVERVIEW

The I2P network is an anonymous overlay network started in 2003, that provides strong privacy protections for communication over the Internet. I2P uses onion routing to hide the location of clients and servers, by sending packets through tunnels that are identified by Destinations (privacy-preserving IP addresses). Routing information for both the public routers themselves and the location-hidden Destinations are stored in a floodfill DHT, which is managed by a dynamic subset of the network participants (floodfill routers). I2P does not have a formal outproxy or exit-node concept within its network architecture; all communications happen within the I2P network.

3. DESIGN CHOICES: I2P VS. TOR

3.1 Packet-switched vs. circuit-switched

Data in the Tor network is routed through circuits, built by the client or onion service that uses them. Connections are allocated to a circuit, and this allocation does not change until either the connection or circuit closes.

By contrast, I2P is an inherently packet-switched network. Routers maintain multiple tunnels per Destination, and can use any of them for any packet. This gives routers significantly more flexibility in their routing strategies (although the majority of routers simply use the reference implementation, which at present uses a relatively simple strategy). From a functional perspective, being packet-switched has several benefits:

- Native support for both TCP-like and UDP-like protocols inside the I2P network.
- Implicit transparent load balancing of messages across multiple peers, rather than a single path.
- Resilience against failures by running multiple tunnels in parallel, and rotating tunnels.
- Each client's connections scale at O(1) instead of O(N): Alice has e.g. 2 inbound tunnels for a Destination that are used by all of the application-layer peers Alice is talking with, rather than a separate circuit for each.

The effects of this choice on I2P's current and potential privacy properties have yet to be formally studied, although no doubt there are such discussions in more general or Tor-related papers that could be applied to I2P. Examples of potentially useful avenues include:

- The effect on request/response identification of running multiple connections over a single tunnel.
- Using multiple tunnels at once for a single connection [10, 5].
- Strategies for marking and batching high-latency packets, to smooth out the bandwidth usage curve [6, 9].

3.2 Unidirectional vs. bidirectional tunnels

Tor clients use interactive telescoping to create circuits through the Tor network: the RELAY EXTEND message [4] is used to progressively lengthen the circuit through the desired hops [2]. Once built, the circuit is used for bidirectional communication; the client sends packets out through the circuit to the server, and return packets are relayed back through the circuit to the client.

I2P takes a significantly different approach: all tunnels are unidirectional. That is, participants in a tunnel only ever send packets in a single direction. In order to support bidirectional communication, routers maintain two "pools" of tunnels (per Destination) - one for inbound packets, and one for outbound. This has several follow-on effects:

- Non-interactive telescoping is used to build tunnels in a single pass. The tunnel builder creates a packet encoding the build request and sends it to the first hop. That hop inserts its reply, and forwards the request on to the next hop, and so on. The final hop then sends the packet (now full of responses) back to the tunnel builder via a specified inbound tunnel. For building inbound tunnels, the process happens in reverse.

- For the default tunnel length of 3 hops, a round-trip communication between two I2P applications will traverse through 12 hops in the network; on average twice as many as for the equivalent setup using Tor onion services, with the obvious resulting latency hit. (Note however that the asymmetry between Tor clients and onion services means that Tor users will end up building more circuits that would otherwise be necessary.)
- As tunnel participants never have a direct communication channel open with the tunnel owner, they cannot inform them of any changes in circumstance. Thus the network-wide 10-minute lifetime of tunnels is intended to provide a balance between tunnel longevity and likelihood of tunnel breakage.
- Tunnel participants see half as much data in unidirectional tunnels, which should in theory make it harder to detect a request/response pattern (as the adversary would need to position themselves in both outbound and inbound tunnels to gain the full picture).

Outside of a brief analysis in a previous PETS paper [8], there has been (to our knowledge) no study of the complex trade-offs between unidirectional and bidirectional tunnels.

3.3 Peer profiles vs. directory authorities

Tor's relay network is managed by a set of nine Directory Authorities. These servers hold the "keys to the kingdom" - they collectively agree on which relays are members of the global network, and publish the membership set (along with statistics about each relay) in a consensus document. Every Tor client can therefore gain a full view of the entire network from this document, and use it for path selection.

I2P's network information is stored in a DHT, and as such there is no central location to obtain reliable router statistics. No I2P router has a full view of the global network; even the floodfill routers that maintain the DHT do not necessarily know all other floodfills. More crucially, the information in the DHT is inherently untrusted - at most, it can be authenticated to being from a particular router, but an adversary could trivially lie in their published data.

To circumvent this problem, I2P routers create "peer profiles" for all routers they interact with. These profiles measure various statistics, such as estimated bandwidth capacity, how often tunnels built through that peer fail, and so on. The router then uses organises these profiles into "tiers", and builds its tunnels using the fastest and most reliable peers it knows (the "fast tier"). This is similar to Tor's usage of the "Fast" flag [2].

The obvious benefit to using peer profiles is that they are not based on any central measurement or data-collection points, and are therefore harder for an adversary to influence, particularly when the tunnel builder is not known. However, traffic confirmation attacks (where the adversary guesses the router behind a particular Destination and then carries out targeted attacks to confirm their suspicions) can be improved by influencing the performance of peers in the victim's fast tier [8].

If the only available information is the Destination itself, then another aspect comes into play: the strategy by which routers select peers from their fast tier for use in tunnels. By inspecting the published inbound tunnel gateways for a Destination, an adversary can attempt to build up a partial view of the target's fast tier. Several defences have been implemented as a result of previous research [8, 7] to inhibit the usefulness of this avenue:

- The fast tier is split into four sub-tiers based on the DHT keys of the routers. Tunnel participants are selected from these sub-tiers depending on their position within the tunnel. An adversarial peer in the fast tier will therefore only ever be selected for the start, middle, or end of a tunnel. To alter this position, the adversarial peer would need to alter their RouterIdentity, which would mean they are no longer in the fast tier.
- The DHT keys of fast tier members are hashed with a random value that is unique per-Destination, per-direction, before splitting into sub-tiers. This effectively acts as a verifiably-random shuffle, and means that even if an adversary can manage to predict which inbound sub-tier they will end up in (based on observations of the published inbound gateways), they cannot use that information to predict where they will end up in outbound tunnels.

Further research on strategies for both peer profile creation and tunnel peer selection would be very benefitial.

4. REFERENCES

[1] I2P Anonymous Network. `https://geti2p.net`.
[2] Tor Path Specification. `https://spec.torproject.org/path-spec`.
[3] Tor Project: Anonymity Online. `https://www.torproject.org/`.
[4] Tor Protocol Specification. `https://spec.torproject.org/tor-spec`.
[5] M. AlSabah, K. Bauer, T. Elahi, and I. Goldberg. The Path Less Travelled: Overcoming Tor's Bottlenecks with Traffic Splitting. In *Proceedings of the 13th Privacy Enhancing Technologies Symposium (PETS 2013)*, pages 1–20, 2013.
[6] C. Diaz, S. J. Murdoch, and C. Troncoso. Impact of Network Topology on Anonymity and Overhead in Low-Latency Anonymity Networks. In *Proceedings of the 10th Privacy Enhancing Technologies Symposium (PETS 2010)*, 2010.
[7] C. Egger, J. Schlumberger, C. Kruegel, and G. Vigna. Practical Attacks Against the I2P Network. In *Proceedings of the 16th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2013)*, Oct. 2013.
[8] M. Herrmann and C. Grothoff. Privacy Implications of Performance-Based Peer Selection by Onion Routers: A Real-World Case Study using I2P. In *Proceedings of the 11th Privacy Enhancing Technologies Symposium (PETS 2011)*, July 2011.
[9] C. T. Oya and F. Perez-Gonzalez. Do dummies pay off? Limits of dummy traffic protection in anonymous communications. In *Proceedings of the 14th Privacy Enhancing Technologies Symposium (PETS 2014)*, pages 1–20, 2014.
[10] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving datacenter performance and robustness with Multipath TCP. In *Proceedings of ACM Sigcomm*, 2011.