APPENDIX

A. ACTION BOUNDS

We determine a set of action bounds to use during our Tor measurements. As described by Jansen and Johnson [2], these describe the maximum amount of network activity that our privacy techniques will protect (*i.e.*, the amount of activity to which our differential privacy guarantee will apply). These bounds apply to users of the Tor network, which include both Tor clients and onion services.

1 Exit-circuit domains
Activity type: Making a connection to a domain over an exit circuit's initial stream
Activity bound: 20/day
Explanation: A Tor Browser user performs this action with each visit to a new domain (*i.e.*, the name in the browser bar) or using a new tab. We protect 20 such actions per day, as this corresponds to 2-4 such actions for 5-10 hours of the day. Note that page loads to the same domain within the same tab (e.g. clicking intra-site links) will go over the same circuit and thus don't count toward this bound.

2 Loading webpage (intermediate action)
Activity type: Loading a Web page (this activity is not directly observable and is just used to inform bounds on directly-observable Tor actions)
Activity bound: 200/day
Explanation: A Tor Browser performs this action when a URL is entered into the browser bar or when a link is clicked. A limit of 200/day protects 20-40 such actions per hour for 5-10 hours a day. Note that this bound is consistent with the limit of 20 initial-stream connections, as for each of those connections this bound allows at least 10 page loads sharing that circuit.

3 Exit data
Activity type: Transferring data over exit connections
Activity bound: 400 MB/day
Explanation: According to http://httparchive.org/interesting.php, the median total transfer size on the Web is at most 2MB. For a Tor Browser user making many page loads, it seems reasonable to limit the average data transferred during each such load to at most this size. The number of page loads we protect is 200/day, and so the number of bytes that we will protect is 200*2MB = 400MB.

4 IP Addresses
Activity type: Connecting via a different public IP address to a Tor relay
Activity bound: 4/day for a single day, 2/day for multiple days
Explanation: A client may change locations a few times in a given a day, and each location may have a different public IP address. For example, a mobile device or laptop may be used on home wifi, work wifi, on shared public wifi, and on a mobile network in the same day. However, we expect that this worst-case behavior isn't reasonably sustained for consecutive days, and so we reduce the average per day from 4 to 2 when considering multiple consecutive days.

5 TCP connections
Activity type: Making a TCP connection (aka OR connection) to a Tor relay
Activity bound: 12/day
Explanation: A TCP connection is created by a client only once during the time that Tor Browser is running. The connection is to the client's (single) guard, and all Tor circuits are multiplexed over that connection. 12 connections/day allows Tor Browser to be opened and closed, or change IP addresses 12 times during the day.

6 Entry circuits
Activity type: Making a circuit via a Tor entry relay (Guard)
Activity bound: 651/day
Explanation: We will protect activity that would arise from a client, but note that this activity may be observed on the (onion) server side as well as the client side. We want to protect the maximum number of circuits created by an exit client, or v2 or v3 onion service client. We also protect additional circuits for directory fetches.

- For exit clients, we assume that a circuit is created for every exit-circuit domain: 20/day
- For onion service clients, we assume that a circuit is created for every descriptor lookup, introduction point connection, and rendezvous connection:
  v2 Tor Browser: 30 descriptor + 30 introduce + 30 rendezvous = 90 circuits/day
  v3 Tor Browser: 30 descriptor + 30 introduce + 30 rendezvous = 90 circuits/day
- For Ricochet, we combine both the onion service and onion service client bounds, because Ricochet uses both. But Ricochet clients only ever have one onion address, so their service bounds are reduced:
  v2 Ricochet: (30 + 150) descriptor + (90 + 72) introduce + 180 rendezvous = 522 circuits/day
  v3 Ricochet-like: (30 + 256) descriptor + (90 + 72) introduce + 180 rendezvous = 628 circuits/day
- We also protect additional circuits for directory fetches. A client fetches a consensus, and then fetches any new microdescs in that consensus. These requests are split over its 3 directory guards. (1 +

10 hours usage / 1.5 hours per consensus) * 3 directory guards = 23 circuts/day additional

7 Entry data
Activity type: Transferring data via a Tor entry relay (Guard)
Activity bound: 407 MB/day aka 407 megabytes/day aka 4.07e8 bytes/day
Explanation: We re-use the action bounds for Exit data and Rendezvous data, because exit and onion clients primarily download data. We assume that a client or service's total usage across both Exit and Rendezvous is less than these bounds. But clients also download compressed microdesc consensuses every 1.5 hours, and microdescriptors when they first start up. (After initial bootstrap, daily microdescriptor downloads are negligible.) We re-use the 5-10 hours per day client usage. So we protect an *additional*: (0.6 MB compressed microdesc consensus) * (1 + 10 hours usage / 1.5 hours per consensus) + (1.8 MB compressed microdescs) = 7 MB/day

8a Descriptor upload (v2)
Activity type: Uploading a version 2 onion-service descriptor
Activity bound: 450/day
Explanation: This bound is the maximum number of descriptor uploads that a v2 onion service would perform over one day: (3 onion addresses) * (1 upload / onion address / HSDir / hour) * (6 HSDirs) * (24 hours/day) * (25 / 24 overlap) = 450 uploads/day. By default, v2 onion services upload descriptors every 0-120 minutes. Services also re-post their descriptor when their intro points expire or fail. We disregard uploads to the same HSDirs during the 1 hour overlap: the collection period is 24 hours, but each service uploads to the same HSDirs for 25 hours, so at least 1 hour of overlap falls outside the collection period. (We account for uploads to the next set of HSDirs during the overlap period by multiplying by 25 / 24.)

8b Descriptor upload (v3 - unused)
Activity type: Uploading a version 3 onion-service descriptor
Activity bound: 768/day
Explanation: This bound is the maximum number of descriptor uploads that a v3 onion service would perform over one day: (3 onion addresses) * (90 / 60 upload / onion address / HSDir / hour) * (8 HSDirs) * (24 hours/day) * (48 / 24 overlap) = 768 uploads/day. By default, v3 onion services upload descriptors every 60-120 minutes. We disregard uploads to the same HSDirs during the 24 hour overlap period: the collection period is 24 hours, but each service uploads to the same HSDirs for 48 hours, so at least 24 hours of overlap fall outside the collection period. (We account for uploads to the next set of HSDirs during the overlap period by multiplying by 48 / 24.)

8c Unique-address descriptor upload
Activity type: Uploading descriptors for unique onion addresses
Activity bound: 3/day
Explanation: We would like to protect an onionsite that uses 3 different onion addresses (say, one for CDN resources).

9a Descriptor lookup (v2)
Activity type: Looking up a version 2 onion-service descriptor
Activity bound: 30/day
Explanation:

- A Tor Browser user visiting an onionsite for the first time will look up its onion address and any different onion address contained that hosts a page resource. We allow that 25% of the onionsites behind the initial domain use two additional addresses (as in bound #8c). v2 onion-service descriptors will be cached and reused on later visits, for approximately 3 days. Re-using the exit assumption of 5-10 hours of browsing per day, we assume that all these descriptors will remain in the cache for the entire browsing session. We protect 20 domains/day, and so we protect: 20 domains/day * (1 site / domain + 0.25 * 2 sites / domain) = 30 descriptor lookups/day.
- A Ricochet user looks up an onion service for each of their contacts regularly, to maintain presence information. Some Ricochet users have large numbers of contacts. We think a reasonable bound for online contacts is the minimum Ricochet window size, which holds 15 contacts. We also allow the same number of offline contacts. So we protect: 15 online contacts/day + 15 offline contacts/day = 30 descriptor lookups/day.

9b Descriptor lookup (v3 - unused)
Activity type: Looking up a version 3 onion-service descriptor
Activity bound: 30/day
Explanation: We use a similar argument to v2 descriptor lookups. v3 onion-service descriptors will be cached and reused on later visits, for approximately 13.5 hours. Re-using the exit assumption of 5-10 hours of browsing per day, we assume that all these descriptors will remain in the cache for the entire browsing or Ricochet session. So we protect the same number of v3 descriptor lookups as v2 descriptor lookups.

9c Unique-address descriptor lookup
Activity type: Looking up a descriptors for different onion addresses

Activity bound: 30/day
Explanation: Using the previous bound on a Tor Browser user visiting 20 unique domains/day, we allow that 25% of the onionsites behind the initial domain use two additional addresses (as in bound #8c): 20 domains/day * (1 site / domain + 0.25 * 2 sites / domain) = 30 unique addresses/day.

10 Introduction Point (IP) connections
Activity type: Making an Introduction Point (IP) connection
Activity bound: 216/day
Explanation: We would like to protect both clients and servers making connections to an IP.

- A client makes 30 connections/day (reusing the rendezvous point argument).
- A service makes 6 IP circuits per onion address, then re-uses them for many introductions, but reconnects during the day. (3 onion addresses) * (6 service introduce connections / reconnect) * (12 reconnections/day) = 216 introduction connections /day.
- But a Ricochet user connects to an onion service for each of their contacts regularly, to maintain presence information. And their client is also an onion service, which uses 6 introduction points. We protect 15 client connections (reusing the online contact limit given for descriptors). But since the Ricochet protocol is bidirectional, we only expect one connection attempt in any direction, per online contact, therefore we assume only half the online contacts use the client side. We assume there are 6 service connections. And all connections reconnect up to 12 times/day (reusing the OR connection limit). So we protect: (15/2 client + 6 service introduce connections / reconnect) * (12 reconnections/day) = 162 introduction connections/day.

11 Rendezvous Point (RP) connection
Activity type: Making a Rendezvous Point (RP) connection
Activity bound: 180/day
Explanation: We will protect activity that would arise from a client, but note that this activity may be observed on the (onion) server side as well as the client side.

- Visiting onionsites at the bound of 20/day (reusing the page-load limit given for clients (#1)) with up to 50% more onion addresses than pages (reusing the argument from unique descriptors (#8c)) would generate 30 RP connections/day.
- A Ricochet user connects to an onion service for each of their contacts regularly, to maintain presence information. And their client is also an onion service. But since the Ricochet protocol is bidirectional, we only expect one connection attempt in any direction, per online contact. So we protect 15 connections (reusing the online contact limit given for descriptors), reconnecting up to 12 times/day (reusing the OR connection limit). So we protect: (15 rendezvous connections / reconnect) * (12 reconnections/day) = 180 rendezvous connections/day.

12 Rendezvous data
Activity type: Transferring data over rendezvous connections
Activity bound: 400 MB/day or 803213 cells/day
Explanation: We would like to protect both client and servers transferring data over an RP. We will protect the higher of reasonable client and server activity.

- A limit on RP bytes per client is 400 MB/day by the same argument as applied to transferring data over exit connections.
- A limit on RP bytes per service uses the previous limit on RP connections per day as well as the argument that, over many Web connections, a reasonable limit on average page size is 2MB. This yields a limit of (2 MB / page load) * (200 page loads/day) = 400MB/day, which is the same limit obtained for clients. But rendezvous points can only see cells. Each relay cell has a 498-byte data payload [1]. Therefore, the action bound is: 400 MB/day / 498 data bytes per cell = 803213 cells/day (rounded up).
- We expect the volume of data transferred over interactive protocols like Ricochet to be minimal.

2. REFERENCES

[1] Tor Protocol Specification. https://gitweb.torproject.org/torspec.git/tree/tor-spec.txt.
[2] Rob Jansen and Aaron Johnson. Safely measuring tor. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1553–1567. ACM, 2016.