A Forensically Sound Method of Identifying Downloaders and Uploaders in Freenet

Brian N. Levine
Univ. of Massachusetts Amherst
MA, USA

Marc Liberatore
Univ. of Massachusetts Amherst
MA, USA

Brian Lynn
Univ. of Massachusetts Amherst
MA, USA

Matthew Wright
Rochester Institute of Technology
NY, USA

ABSTRACT

The creation and distribution of child sexual abuse materials (CSAM) involves a continuing violation of the victims' privacy beyond the original harms they document. A large volume of these materials is distributed via the Freenet anonymity network: in our observations, nearly one third of requests on Freenet were for known CSAM. In this paper, we propose and evaluate a novel approach for investigating these violations of exploited childrens' privacy. Our forensic method distinguishes whether or not a neighboring peer is the actual uploader or downloader of a file or merely a relayer. Our method requires analysis of the traffic sent to a single, passive node only. We evaluate our method extensively. Our in situ measurements of actual CSAM requests show an FPR of 0.002 ± 0.003 *for identifying downloaders. And we show an FPR of* 0.009 ± 0.018*, a precision of* 1.00 ± 0.01*, and a TPR of* 0.44 ± 0.01 *for identifying uploaders based on in situ tests. Further, we derive expressions for the FPR and Power of our hypothesis test; perform simulations of single and concurrent downloaders; and characterize the Freenet network to inform parameter selection. We were participants in several United States Federal Court cases in which the use of our method was uniformly upheld.*

CCS CONCEPTS

• **Security and privacy** → **Privacy protections**; • **Applied computing** → **Network forensics**.

KEYWORDS

forensics; child sexual abuse materials; child rescue; darknets; anonymous communication systems

ACM Reference Format:
Brian N. Levine, Marc Liberatore, Brian Lynn, and Matthew Wright. 2020. A Forensically Sound Method of Identifying Downloaders and Uploaders in Freenet. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20), November 9–13, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3372297.3417876

CCS '20, November 9–13, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7089-9/20/11…$15.00
https://doi.org/10.1145/3372297.3417876

1 INTRODUCTION

Anonymous communication systems, including Freenet [1, 2] and Tor [3], play a critical role in the sexual abuse of children [4–26]. They are more than venues and conduits for the trafficking of *child sexual abuse material* (CSAM[1]). They mask the perpetrators' activities and they ensure persistence of CSAM over many years [27], which is a continued victimization of depicted individuals into their adulthood [28–32]. CSAM made available via anonymous systems is used by perpetrators to groom new victims through normalization of the acts they depict [33]. These acts represent the gravest privacy violations of the child victims. Therefore, it is critical that investigators have available to them methods to stop these crimes on anonymous communication systems.

Although not as popular as Tor and often overshadowed by it, Freenet has been consistently used by thousands of users a day for decades. Files published to Freenet are fragmented into small encrypted *blocks* that are dispersed randomly throughout the network of peers. A *manifest key* is a URI necessary to retrieve and reconstruct the original file. Manifest keys for published files are often broadcast via open, public forums. In addition to representing files on the network, manifest keys are used to find HTML-based websites called *freesites* and a Usenet-like forum called *Frost*.

We found a number of user-created areas on Freenet explicitly and openly dedicated to child sexual exploitation. We harvested over 124,000 manifest keys from these public areas. Law enforcement was able to confirm that many referenced CSAM images; and half of those they identified were deemed *severe*, meaning they involved very young victims or violent acts. We then took a sample of Freenet traffic, and found that over 30% of all requests on Freenet were for the keys we harvested.

To stop these crimes, investigators require a method that is both effective and *forensically sound*. We adopt the *Daubert* standard [34] as our definition of forensic soundness. That is, we seek a method that is based on a testable hypothesis, has a known error rate, follows existing standards, and uses generally accepted methods. Prior work has revealed some vulnerabilities in Freenet's approach [35–40] that could in theory be applied to forensic investigations. However, these past works are relatively heavyweight approaches that would require active maneuvering of the investigator's position, the use of active traffic probing, the use of multiple peers, or relied on ephemeral bugs. Furthermore, Freenet developers have largely addressed these vulnerabilities.

[1] CSAM is often called *child pornography.*

Contributions. In this paper, we design and evaluate a method for investigation of CSAM-based privacy violations and re-victimization on Freenet. We present a novel approach to distinguishing whether or not a neighboring peer is the actual uploader or downloader of a file in Freenet. Our algorithm can be enacted by a single peer using only the passive analysis of the traffic that is sent to it by the neighbor. Our method is strongly justified, and we show it to be effective for investigators and forensically sound. Our efforts place a priority on quantifying and eliminating possible false positives. We also detail use of the method to rescue children from abuse. Our method has been considered by several US District Courts, and all have upheld its use in criminal investigations. Our technique is based on a simple observation of Freenet: requests associated with a manifest are divided among neighbors, and counts of requests fall exponentially with each hop from the original uploader or downloader. Our specific contributions are as follows.

- **Motivation.** We observe that at least 30% of request traffic on Freenet is for CSAM-related content. This fraction is consistent over four years of measurements that we have made. We also analyze, for the first time, the basis of Freenet's anonymity mechanism and prove it prevents the use of hop counters for de-anonymization.
- **Investigative technique and in situ evaluation.** We derive an investigative technique based on a Bayesian hypothesis test that considers request traffic. The test models the investigator's decision for whether or not a neighboring subject is a downloader or uploader of content. To evaluate the technique, we placed nodes on the real network and waited for them to be used as relays for actual CSAM investigated by law enforcement; we observed an FPR of 0.002 ± 0.003 ($n = 918$). In separate tests, we uploaded non-CSAM content to the real network. We observed an FPR of 0.009 ± 0.019 ($n = 111$), a precision of 1.00 ± 0.01 ($n = 4,415$), and a TPR of 0.44 ± 0.01 ($n = 9,965$) for our technique.
- **Analysis and simulation.** We derive expressions for the FPR and *Power* of our Bayesian test. The expressions validate our in situ evaluation, predicting that for typical scenarios on Freenet, the probability of false positives is very low and decreases with the size of the file shared. Meanwhile, the probability of a false negative remains low. This result is further validated by simulations where we construct and vary the topology. Additionally, we show through simulation why the traffic of other, concurrent downloaders that might pass through the subject is not a significant source of false positives. We detail and validate assumptions of our model and set parameters based on Freenet's design, its operation, and characterization of its network.
- **Deployment and legal outcomes.** We disclosed our work publicly [14]. We were consulted during investigations that made use of the method, and we were participants in several US Federal Court cases. We summarize the outcomes, which resulted in justice for re-victimized persons whose depictions of abuse were in possession by the downloader, and in one case, the rescue of children from hands-on abuse by a Freenet downloader.
- **Anonymous system design.** We discuss user and developer changes to Freenet that may allow requesters to avoid detection.

We conclude by comparing to **related work** and reviewing **ethical concerns**. Given these results, we argue that this investigative technique is a highly accurate, efficient, effective, and forensically sound method of addressing child privacy violations resulting from the use of Freenet to upload and download CSAM.

This paper expands significantly on a prior workshop publication by the authors [14] and is distinguished by the following new contributions: a proof that de-anonymization cannot be based on Freenet's probabilistic packet hop counters (Theorem 1 in §4.2); parameter validation based on characterization of Freenet's topology (§4.4); an empirical measurement of FPR for downloaders based on years of in situ experiments (§5.1); expansion of our method to identification of uploaders, including an empirical measurement of precision, TPR, and FPR based on in situ experiments (§5.2); an analytical model of FPR and Power (§5.3: Theorems 2, 3, 4, and 5 and Lemmas 1 and 2); simulation-based evaluation of the false positives caused by concurrent downloads (§6.2); and reporting of outcomes from use of the method by law enforcement in practice (§7). Overall, each of these new results adds substantially to our assessment of the forensic soundness of the technique by demonstrating via multiple methodologies that it produces reliable evidence.

2 BACKGROUND

Freenet is an overlay network that operates as a distributed data store, with each participating *node* anonymously contributing key-value storage to encrypted *blocks* of data [1, 2, 41]. A key is the SHA256 hash of the block. Freenet nodes form a small-world network [42] with each node connecting to other nodes, the set of which we call its *peers* or *neighbors*. Unlike anonymity systems, Freenet users do not attempt to hide their IP addresses behind proxy nodes (e.g., Tor guard nodes). Instead, nodes attempt to blend their traffic into traffic generated by other nodes. Freenet has two operational modes: *darknet* and *opennet*. In darknet mode, Freenet nodes only connect to peers for which the user has explicitly given permission. In opennet, nodes connect to other opennet nodes, discovered from well-known seed nodes or other nodes. Opennet allows for neighbors to exchange information on their peers and form new connections to better organize the network. We focus exclusively on opennet. Our method should work for darknet, though it would also require social engineering to join specific darknets.

Uploading Files and Manifests. For a file to become available to Freenet nodes, it must first be uploaded (or inserted) by a node into the network. Nodes do not explicitly share files; files must be retrieved from the network's distributed data store. When a file is inserted into the network, Freenet divides it into encrypted 32KB *content blocks* (or simply *blocks*). The inserting node (or *requester*) distributes the blocks to its peers. A peer may place the block in its own storage, and it may also send the block to its own peers.

After the requester distributes the content blocks, it inserts a separately encrypted 32KB *manifest block*. The manifest block holds the SHA256 hash and decryption key of each of the file's content blocks. For large files with many blocks, the manifest block cannot reference all of the content blocks and will instead reference another level of manifest blocks. Finally, Freenet returns a human-readable *manifest key* to the user. The manifest key is a URI, consisting of the manifest block's SHA256 hash and decryption key. Freenet allows a user to decide between two primary types of manifest key: content hash key (CHK) or signed subspace key (SSK). CHK blocks will be identical for multiple copies of the same file. SSKs generate a unique encryption key per upload, and thus the blocks would be distinct per manifest and uploader. Anyone with knowledge of the manifest key can retrieve the file.

Downloading. Retrieving a file from the network is the inverse of inserting it. The downloading node—also known as the *requester*—first retrieves and decrypts the manifest block(s) to get the hash and encryption keys of the content blocks. The requester then retrieves and decrypts the content blocks, and finally reconstructs the file data. Downloading a file does not result in the file's blocks being placed into the node's local storage.

In requesting blocks, a simple process is followed. If a node receiving a request has the block in its Freenet storage, it returns the block. Otherwise, it relays the request to one of its peers. The process continues from peer to peer. When found, the block is returned in reverse order through the chain of peers, and cached by each for faster replies to future requests. If a node in the chain fails to find the block, it may relay the request to other peers before returning a not-found result.

Peer Selection and Routing. Every Freenet node and block is assigned a *location*, a 64-bit floating point number between 0 and 1, inclusive. Locations are points on a circular ID space, with 0 and 1 being the same point. The *distance* between any two locations is the length of the shortest arc between the two points. A persistent location is randomly assigned to each opennet node, and each block's SHA256 hash can be deterministically converted to a location.

An opennet node selects the majority of its peers from among nodes close to its own location, with the remaining peers distributed throughout the circle. The total number of peers is a function of the upload bandwidth the user allocates to the Freeent node. A node sends a request in the direction of the node closest to the block's location. Freenet performs *friend-of-a-friend* (FOAF) routing: nodes have visibility to their immediate peers' locations, as well as the locations of their peers' peers. All visible locations are considered when selecting a recipient peer. If a request is unsuccessful, the next closest peer is tried, once again taking FOAF routing into account. A peer may choose to reject a request due to resource constraints. Receiving rejects may cause a node to mark its peer as being *backed off*, temporarily removing the peer as a routing option.

At any given instant, some peers are responsible for larger portions of the ID space than others. FOAF routing results in requests being distributed more proportionally to peers, rather than simply reflective of the amount of space a peer occupies on the circle. Over time, as peers come and go, the distribution becomes more equal.

Hops-to-Live. Freenet uses a *hops-to-live* (HTL) counter to prevent block requests from propagating indefinitely. Generally, the HTL begins at 18 and is decremented by each relay node until it is zero, in which case a not-found error is returned (if downloading).

A requester would be revealed if Freenet were to always start with an HTL of 18. To anonymize the requester, a requesting node will randomly choose, with probability 0.5, whether to use 18 or 17 before originating a request to a peer. Once an initial HTL value is selected, it is permanent for all originating requests sent to that peer. The decision also applies when relaying requests received with an HTL of 18. Insert requests have some variations in behavior (see §4.4). Regardless, the HTL does not reveal the originator. For clarity, we describe requests with an HTL of 18 or 17 as potentially being from a requester, and 16 or below as not being from a requester. However, our technique would work even if other initial HTL values were selected by a user.

Date	Percent
December 2016 – January 2017	34.5%
December 2017 – January 2018	38.7%
December 2018 – January 2019	31.8%
December 2019 – January 2020	30.6%

Table 1: Requests related to CSAM. HTL 18 and 17 only.

Data and FEC Blocks. The number of content blocks Freenet inserts into the network is considerably more than what one would expect by simply dividing the file's data into 32KB blocks. In addition to the *data blocks*, Freenet uses Reed-Solomon codes to generate forward error correction (FEC) blocks. These *FEC blocks* provide redundancy and the ability to recreate missing blocks. Freenet subdivides a file into *segments*. For each n data blocks in a segment, Freenet inserts $n + 1$ FEC blocks. A segment may not reference more than 256 total blocks. To recreate the file data represented by a segment, Freenet must successfully fetch n content blocks, using any combination of data or FEC blocks. If Freenet fails to fetch a block, it will randomly select another block from the segment to fetch. Freenet can fail on half of the blocks, yet still succeed in its download.

Freenet attempts to increase the availability of recently requested files. Once a segment has been reassembled, the requesting node will "heal" the file by regenerating then inserting each block in the segment that it had requested but failed to fetch. To further increase block availability, a node that is downloading blocks may randomly re-insert a fetched block, with each block having a 0.5% probability of being selected for re-insertion.

3 MEASUREMENT OF CSAM ON FREENET

We harvested more than 124,000 manifest keys posted to freesites and Frost boards explicitly dedicated to child exploitation, and Freenet was queried for the block keys associated with those manifests, resulting in the collection of more than 300 million distinct keys. We did not download the files.

A file's SHA1 hash is often available as metadata in the file's manifest block. We were able to identify the SHA1 file hashes for approximately 68,000 of the harvested manifests. Law enforcement confirmed that about 22,000 of the SHA1 hashes were known to them as CSAM, and approximately 14,000 of those were identified as being *severe*. Severe CSAM is typically defined to either have very young victims or involve violent acts against the victims. We believe that the files whose SHA1 values were not known stand a good chance of being CSAM; this was confirmed by law enforcement visually inspecting some of the unknown files.

Measurements. We operated a minimum of ten Freenet opennet nodes from November 2016 through April 2020, inclusive. Past work has measured Freenet extensively, including active probes from Roos et al. [39, 43]. We did not probe the network or modify the software except to log friend-of-a-friend routing information sent to our nodes, and to log requests sent to our nodes for keys that matched those that we harvested. We also logged a count of the block requests sent to our nodes having an HTL of 18 or 17, regardless of whether the keys had been identified.

During December 2019 and January 2020, 31% of such requests were for those keys that we harvested; see Table 1. A consistently high percentage of Freenet requests are related to these manifests: we observed 32% during the same period in the year before, and 39% and 35% for the two years prior. Over 250 million block requests were observed during each period. As noted, law enforcement identified more than 22,000 files in these 124,000 manifests, and though some remain unconfirmed, it is also unlikely that we have harvested all requested CSAM manifests. In sum, it is reasonable to assume that a minimum of 30% of Freenet's traffic is related to child exploitation material.

We also seek to estimate the number of distinct Freenet users. Such estimates are difficult to come by, as prior longitudinal studies have been discontinued [44]. In September 2019, we began running an additional 20 opennet nodes to perform our upload experiments. These nodes were modified to log: *(i)* Freenet neighbor and FOAF locations reported by their peers, *(ii)* data and insert requests received from their neighbors, and *(iii)* data and insert requests initiated or relayed by those nodes. During March 2020, 16,971 distinct opennet peer locations were visible to our nodes, with an average of 4,607 distinct locations per day. Tor and BitTorrent are reported to have many more users [27, 45], but because Freenet is heavily used for CSAM trafficking, there would be motivation for law enforcement to focus investigations on Freenet.

4 INVESTIGATIVE TECHNIQUE

The goal of the investigator is to identify whether a neighbor is the *requester* node, i.e. the downloader or uploader of a file, or is instead a *relayer* node, i.e. forwarding requests from others. In this section, we develop an investigative model to distinguish between the two scenarios. The HTL of a request, whether 18 or 17, does not indicate that the sender is a requester or relayer. Freenet's source code makes this assertion,[2] but it was never proven formally; we provide a proof below. Thus, another approach is required.

Our method is based on a simple observation. In our model, an observer who is one of g neighbors of the actual requester can expect to receive about $\frac{1}{g}$ of all requests, due to FOAF routing. Block locations, derived from a hash, are modeled as being evenly distributed on the circle. If the observer is actually two hops away from the original requester, then only about $\frac{1}{gh}$ of the requests will be received, assuming the requester has h neighbors. Since the manifest key is overt, an observing peer can determine the number of insert or download requests expected. Accordingly, given the number of requests received, the observer can quantify the probability of whether the requests were relayed by or originated with its neighbor.

4.1 Assumptions and Model

We assume the investigator has collected manifest keys for a limited set of *files of interest*, which are, for example, openly published on Freenet freesites and forums related to CSAM. Freenet provides a mechanism to harvest block hashes given a manifest key. Keys can be logged before or after manifests are published publicly.

For simplicity, we assume the investigator operates a single peer in the Freenet network, which we call the *observer*. Running multiple independent nodes is possible and efficiently allows for parallel investigations. The observer is strictly passive: it participates in the network by forwarding requests as usual, but also logs the SHA256 hash keys of any request it sees, together with the HTL, location of the sending peer, and the count of that peer's neighbors. Surprisingly, these are all the steps required.

4.2 Description

For each manifest key from the files of interest, the investigator fetches the manifest blocks and obtains the keys of each file's content blocks. The observer node passively logs insert and download block requests from its peers. The observer can easily map the requests to the files of interest by the key values. Requests that don't map are not retained. The observer then counts the requests received on a per-peer, per-HTL, and per-file basis, for all known files of interest. Based on the counts, the observer can calculate the likelihood that a given peer is either:

- the actual requester, who is requesting to download or upload blocks for a specific file; or,
- a relayer for the actual requester.

We call the observer's peer who sends the requests the *subject*, as they may be in either role. Figure 1 illustrates the two scenarios.

- Let $H1$ be the hypothesis that the subject is the actual requester.
- Let $H2$ be the hypothesis that the subject is a relayer and a peer of the actual requester.

Assuming the default maximum HTL of 18, requests that originated at the subject would only have an HTL of 18 or 17; we do not need to count requests with lower HTLs. For the remainder of this section, we consider only requests with HTLs of 18 or 17.

HTLs of 18 and 17 provide equivalent information. Freenet's original design [1] used a flawed anonymity mechanism. More than a decade ago, Freenet's codebase introduced a new approach: requests with the maximum HTL value (of 18) are probabilistically decremented [36]. Does receiving requests with an HTL of 18 versus 17 affect the probability that the subject is the original requester? Let $Pr(H1|Y)$ and $Pr(H2|Y)$ denote the probability that $H1$ or $H2$ are true, respectively, given a run of requests with HTL of Y, where $Y = 18$ or $Y = 17$.

> **THEOREM 1:** *The probability that hypothesis $H1$ is true is no different when a run is composed of requests with HTLs of 18 or 17, regardless of the distance of the requester from the observer, if the probability is 1/2 that an 18 is decremented; i.e.,*
>
> $$Pr(H1 \textit{ is true } |Y = 17) = Pr(H1 \textit{ is true } |Y = 18).$$

The proof appears in Appendix A. Theorem 1 implies a different approach is required to distinguish a relayer from an actual requester.

Our approach. The intuition for the investigative technique is now easy to describe. For large files, the actual requester will make a large number of requests for blocks from the manifest, and those requests will be spread randomly among its peers. An observer

[2] https://github.com/freenet/fred/blob/build01475/src/freenet/node/PeerNode.java#L1603

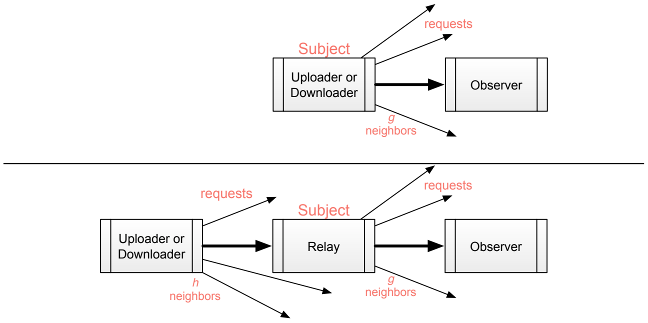


Figure 1: The observer's goal is to distinguish between these two scenarios.

who is a peer of the actual requester will expect to see a certain number of those requests, with some variance. On the other hand, if the observer is a peer of a relayer, it will see only a fraction of the requests seen by the subject. The investigative technique uses a statistical test to distinguish between these two cases.

4.3 Analysis

We now more formally describe our investigative technique, which uses several values as input. Two directly observed per-file, per-peer values are:

- g : the number of directly connected peers of the subject (including the observer herself); and
- r : the number of requests received from the subject by the observer.

The observer must select two additional values:

- T : the total number of requests made by the requester; and
- h : the number of peers assumed connected to a hypothesized source that is not the directly observed subject.

The observer learns g from the Freenet protocol. How T is selected depends on whether the observer is testing for downloads or uploads. Due to redundant forward error correction blocks, only about half of the total defined blocks are required to download a file. The number of download requests made is dependent on the number of blocks available. In practice, we use a downwardly adjusted value of r for downloaders; the number of requests observed represents an upper bound. We define the values used for T and h, as well as detail when and how we reduce r, in Section 4.4.

We construct a model by assuming that each request made by the actual uploader or downloader is sent to exactly one of its peers, and that the selection of that peer is made uniformly at random. The total number of requests an observer will receive if the subject is the actual requester can be modeled by a binomial distribution. Let p be the probability of each request being sent to the observer. Let $X \sim \text{Binom}(T, p)$ be a random variable representing the number of requests received by the observer. Given T possible requests, the probability of the observer receiving $X = r$ requests is

$$Pr(X = r) = \text{Binom}(r; T, p) = \binom{T}{r} p^r (1-p)^{T-r}.$$

In the case of $H1$, the hypothesis that the subject is the actual requester, $p = 1/g$. For $H2$, the hypothesis that the subject is a relayer and a peer one or more hops from the actual requester, we let $p = \frac{1}{gh}$. (In cases where the actual requester is two or more hops away, p could be modeled with a smaller value; however, this approach is sufficient to distinguish the requester.)

As stated above, the technique assumes that the subject is either the actual requester or a relayer directly connected to him. Using a Bayesian framework [46], this assumption can be modeled as follows. We seek the probability of $H1$ given that the observer has received $X = r$ requests.

$$\begin{aligned} Pr(H1|X{=}r) &= \frac{Pr(H1)Pr(X{=}r|H1)}{Pr(X{=}r)} \\ &= \frac{Pr(H1)Pr(X{=}r|H1)}{Pr(H1)Pr(X{=}r|H1) + Pr(H2)Pr(X{=}r|H2)}. \end{aligned} \tag{1}$$

We know that

$$Pr(X = r|H1) = \text{Binom}\,(r; T, {}^1\!/_g)\,,$$

and similarly,

$$Pr(X = r|H2) = \text{Binom}\,(r; T, {}^1\!/_{gh})\,.$$

To set the priors $Pr(H1)$ and $Pr(H2)$, we use the number of peers of the subject as a guide. Assuming that among the subject or his peers, each is equally likely to be the actual requester, we get $Pr(H1) = \frac{1}{g+1}$ and $Pr(H2) = \frac{g}{g+1}$; we discuss our choice of priors further in Section 4.5. Altogether, we have

$$Pr(H1|X = r) = \frac{\frac{1}{g+1}\text{Binom}\,(r; T, {}^1\!/_g)}{\frac{1}{g+1}\text{Binom}\,(r; T, {}^1\!/_g) + \frac{g}{g+1}\text{Binom}\,(r; T, {}^1\!/_{gh})}, \tag{2}$$

where $Pr(H2|X = r)$ is the complement of the above. Eq. 2 represents a standard hypothesis test [46]. One could compare $Pr(H1|X = r)$ to $Pr(H2|X = r)$ and select the hypothesis with the greater probability. However, our goal is to reduce false positives at the expense of false negatives, and we use a higher standard of selecting H1 only if $Pr(H1|X = r) > t$ for some threshold $1/2 < t < 1$.

This model is straightforward, but there are further considerations. Presently, we address how we modify the values above to account for Freenet's real operation. Throughout this paper, we consider false positives. For example, we evaluate the summed traffic from multiple concurrent relayers in Section 6.2.

4.4 Modifications for Real Freenet Traffic

Recall that Eq. 2 estimates the probability that a given subject is a requester on the basis of: g, the number of peers of the subject, which we can observe directly; r, the (possibly adjusted) number of requests observed; h, the number of peers of a possible third-party downloader, which we estimate; and T, the total number of requests made by the requester, which we also estimate. In this section, we describe how we set these values given Freenet's real operation.

Defining *r*. To apply Eq. 2 to real data, the investigator's node observes and logs requests that are sent to it. Requests for keys can be compared to a table of keys harvested from manifests. Thus, any keys in the table can be mapped to a specific file. Requests contain the key, an HTL, the sender's IP address and Freenet location, and the request type (retrieve or insert). The observer also logs a timestamp and the number of peers of the sender. The log is then analyzed to identify *runs* of requests. To reduce potential false positives, we define a run to be a collection of observations where:

- all observations are of requests for blocks associated with the same manifest;
- all observations are of the same peer, as identified by IP address and Freenet location;
- all requests have a consistent HTL (detailed below);
- a minimum of 20 requests for distinct blocks were observed;
- the duration between requests does not exceed a defined value.

A request may be for a data block, an FEC block, or a manifest block. A *data request* is used to fetch a block; an *insert request* is used to upload a block. A download may generate insert requests (Section 2). We require that at least 20 requests are received before a test is run. It's a static value that performs very well in all our evaluations (Sections 5 and 6). We discuss the parameter further in Section 5. We use a static value because there are practical benefits to minimizing the number of variables and equations that need to be presented to laypersons. Similarly, for the duration between requests, we use a static value that performs well in our evaluations and could be changed in the future with additional experimentation.

Defining r_d and T_d for Downloaders. Recall that a manifest consists of about twice as many blocks as are required by Freenet to recreate the original file. If all blocks were available on the network, only roughly half would be requested. However, additional requests are made if blocks are unavailable, and the requests may be sent to multiple peers. This redundancy can inflate o, the number of distinct data requests observed; and without adjustment it could lead to false positives. Therefore, for a given run, we compute

- o, the number of data requests for *distinct* blocks;
- i, the number of insert requests;
- x, the number of duplicate data requests;

and define r_d as

$$r_d = o - i - \epsilon x. \tag{3}$$

Because an insert request typically indicates an additional data request was required, we decrement the count by the corresponding number. We further reduce the count by a constant multiplier, ϵ, for each key with duplicate observations (or triplicates, etc.). Duplicate data requests represent failed data requests or a concurrent downloader. We would have expected the other blocks within a segment to have been requested before re-requesting a block. Since a relayer might request a distinct block from several of its peers due to not-found errors, this can result in a large number of requests and potentially to false positives. We mitigate by applying a multiplier to the number of duplicates observed. We selected a value of $\epsilon = 3$ because a request will be sent three times before it enters a "cool down" period. Analysis of real data confirmed that this choice was effective in limiting false positives.

To determine a suitable value for T_d, we conducted experiments on Freenet where we inserted our own files into the network and then instrumented a downloader to count the number of distinct blocks it requested. On average, it requested 67% of the total blocks. We chose a value of $T_d = 0.8 * TotalBlocks$.

Defining r_u and T_u for Uploaders. In contrast to downloading, when uploading files we know that the requester must insert each of the manifest, data, and FEC blocks. In practice, a requester may send duplicate insert requests to several of its peers. However, with rare exceptions, a requester will only initiate sending the block's contents to a single peer. To set r_u, we only include insert requests where the insertion of the block's content was also initiated, and we are able to set $T_u = TotalBlocks$.

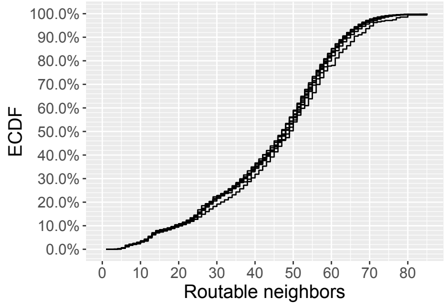


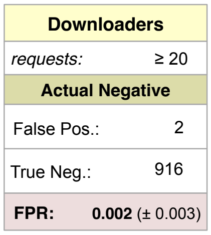
Figure 2: CDF of the count of routable neighbors reported by 21,245 neighbors of our nodes (distinguished by location) from 9/2019–3/2020 inclusive. Each semi-transparent line is one month of data. Overall, 98.0% of reports are for eight or more neighbors, with 99.5% reporting six or more.

Consistent HTLs. In defining runs of r requests, we require consistent HTL values. Data requests are by default sent with either an HTL of 18 or 17, and the same HTL value is used for all requests to the same peer. We require that data requests have HTLs of 18 or 17, but not both. Like data requests, insert requests are initiated with an HTL of 18 or 17, but Freenet may resend the request with a lower HTL value. When testing for uploaders, we require that the run consist of only HTL 18 and 17, or only HTL 17 and 16.

Setting h and g. Our method requires that we set a value for h, the number of peers connected to a hypothesized source. We set the value to $h = 8$. Based on measurements of the network from September 2019 to March 2020 inclusive, this is very conservative in that it reduces the FPR compared to higher values of h, which are more typical. Figure 2 shows an empirical CDF of all reports from 21,245 neighbors, with one line per month. Whenever the Freenet client restarts, it must re-build its set of neighbors, and that is one reason we occasionally observe a small number of neighbors. 98.6% of reports are for eight or more neighbors, with 99.8% reporting six or more. These results are in line with past work [43, 44]. And we note that when we set g, we include neighbors of the subject that are temporarily backed off. These seven months of measurements show that on average 11.2% of the routable peers are backed off at a given moment. Our in situ experiments in Section 5 account for these ephemeral backoffs. The backoff process does not prevent our method from achieving a very low FPR.

4.5 Selecting Priors

In Bayesian statistics, priors are formulated by the experimenter before data is observed [46]. We have followed that approach, selecting well-reasoned priors: $Pr(H1) = \frac{1}{1+g}$ and $Pr(H2) = \frac{g}{g+1}$ These priors are conservative in that they include the possibility that the investigator's node is the requester, which decreases the FPR, especially when g is small. The priors favor H2 by a factor of $({}^{g}/_{g+1})/({}^{1}/_{g+1}) = g$. Regardless, the posterior probability shown in Eq. 2 is the dominant term for typical values of T and r. In idealized scenarios, Bayesian priors can be updated as tests are completed. Individual investigators could update priors by applying the test,



Downloaders	
requests:	≥ 20
Actual Negative	
False Pos.:	2
True Neg.:	916
FPR:	**0.002** (± 0.003)

Figure 3: In situ Freenet downloader experiments. Our nodes were only relays in the download experiments. Downloaders were potentially any other nodes on the network downloading CSAM; thus the FPR is the same as for real investigators.

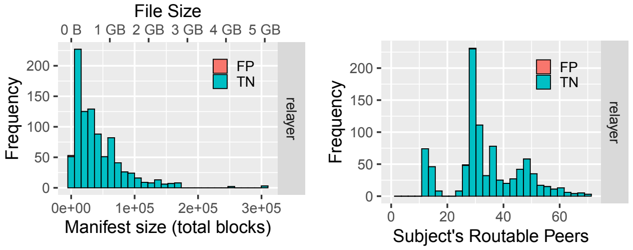


Figure 4: Characteristics of the 918 downloader tests. Two tests were FPs: one with 4,859 blocks and 30 peers, the other with 1,110 blocks and 28 peers.

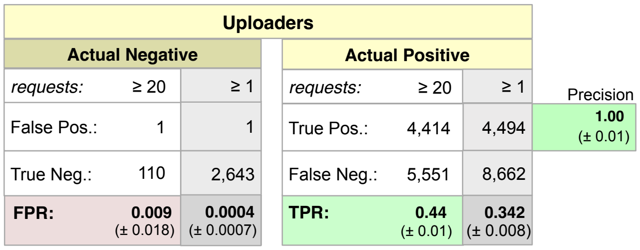
executing the proper legal process, and then carefully measuring the count of targets found with CSAM and those without. Or, our priors could include results from the tests for false positives on the real network that we describe in Section 5. For simplicity, however, we maintain static conservative priors.

5 EVALUATION I: IN SITU TESTING AND ANALYSIS

We use a spectrum of evaluations to show that our method is both accurate and effective. First, we measure the FPR of downloader detection by deploying peers on the network and observing requests for CSAM; thus the FPR is the same experienced by real investigators. We then measure the FPR and TPR of uploader detection on the real network. Second, we derive an analytical model of FPR and *Power* [46] to validate our experiments and provide deeper insights. Our evaluations include the effects of node degree, manifest size, method parameters, and concurrent downloaders. In Section 6, we continue our evaluations using simulation.

5.1 Downloader FPR for Known Negatives

Our first goal is to measure the FPR of our method when it is applied in practice to investigations of CSAM downloading on Freenet. To this end, we deployed many nodes to Freenet that requested no manifests. The experiment began in October 2017 and ran through April 2020. These passive nodes used default installation parameters and were on machines with typical resources. We looked for runs, as defined in Section 4.4, marking the subject as a candidate for our test. We included the requirement that the number of requests must number at least 20. A false positive occurred any time our method flagged the subject as a probable downloader, because none of our subject nodes downloaded any of the over 124,000 CSAM files of interest. We used a threshold probability of $t \geq 0.98$ to determine if a run was associated with a downloader versus a relaying node. Results are shown in Figure 3: we identified 918 runs to be evaluated, and two of the runs were falsely flagged as being from a downloader. In other words, we observe an FPR of $2/918 = 0.002$ with a 95% confidence interval of 0.003.



Uploaders							
Actual Negative			**Actual Positive**				
requests:	≥ 20	≥ 1	*requests:*	≥ 20	≥ 1		Precision
False Pos.:	1	1	True Pos.:	4,414	4,494		**1.00** (± 0.01)
True Neg.:	110	2,643	False Neg.:	5,551	8,662		
FPR:	**0.009** (± 0.018)	**0.0004** (± 0.0007)	**TPR:**	**0.44** (± 0.01)	**0.342** (± 0.008)		

Figure 5: In situ Freenet uploader experiments. Our nodes were the uploaders and sometimes relays, thus we could observe actual positives and negatives. When including runs with less than 20 requests, the FPR is lower and precision remains high. Parentheses show 95% confidence intervals.

The runs of requests relayed through our nodes and triggering our method were for CSAM files that ranged in size, with a distribution shown in Figure 4 (left). The median size was 30,680 blocks (502 MB). Figure 4 (right) shows the number of neighbors that our passive nodes had when they were test subjects. As such, the two false positives were not distinct from the true negatives. (Although difficult to see, the two FPs are included in each histogram.)

Our investigative technique is based on relatively few parameters, whereas Freenet is a real distributed system composed of many moving parts. The outcome of the test may be affected by network size, topology and node degree, third-party traffic, routing and node backoffs, manifest popularity, resource diversity among nodes, and perhaps even undocumented aspects of Freenet code. These 918 tests, completed over a period of years using real CSAM, confirm that our approach is very accurate in practice, despite all the factors that might come into play.

5.2 Uploader FPR and TPR for Known Positives and Negatives

We evaluated the efficacy of our method to identify uploaders in Freenet. We uploaded files containing random bytes, ensuring they were not already in the network. To ensure diversity of topology and neighbors, we deployed 20 Freenet nodes to conduct this experiment, each a default installation with typical resources. Without our influence, occasionally our nodes would become neighbors of one another. We uploaded 15 or 16 manifests from each node of varying size, totaling 309 uploads. For each upload, we logged the number of insert requests received by each neighbor of the uploader. When our own node was an immediate relayer for the uploader, or an immediate neighbor of the uploader's relayer, we logged a count of requests as well. Therefore, we could apply our investigative method as if all neighbors of the uploader and relayer were investigators.

In the case of the uploader, we calculate the TPR (i.e., recall) and precision. TPR is the efficiency of our method in finding actual positives, whereas precision is the fraction of positive results that

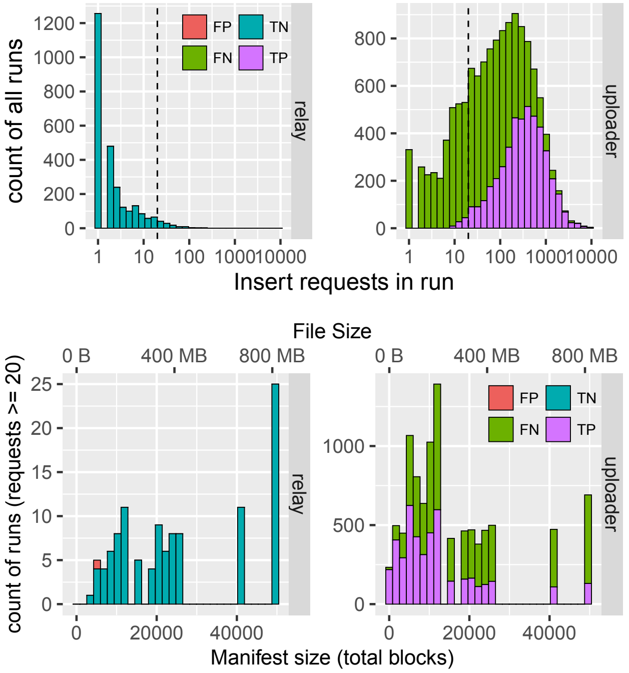


Figure 6: Freenet uploader experiments (semi-logscale for requests). The one FP was for a manifest with 5,127 blocks.

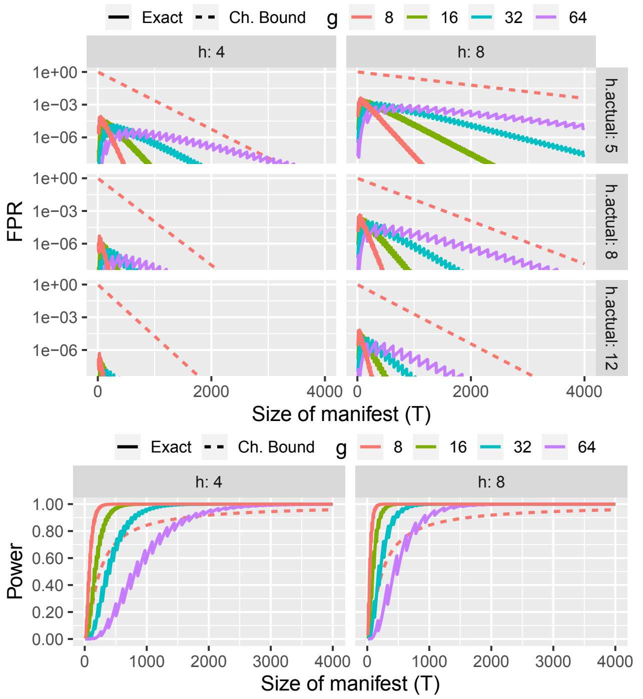


Figure 7: Exact FPR and Power of our method. Calculated from Eqs. 4 and 5. Dotted lines show Chernoff bounds from Eqs. 9 and 17 for $g = 8$.

are true positives. In the case of the relayers of the uploader, we calculate the FPR.

One of our nodes was a relayer or neighbor of a relayer in 2,644 cases total; in 111 of those cases, at least 20 insert requests were sent. We observed one false positive. Figure 5 shows the results: FPRs of 0.0004 ± 0.0007 and 0.009 ± 0.018, respectively, depending on the request threshold. We had thousands of opportunities to calculate the TPR (0.443 ± 0.01) and precision (0.9998 ± 0.01) of our method. Relaxing the requirement of receiving at least 20 requests reduces the TPR, but not precision. As our results show, the minimum could be lowered below 20, but we leave that for future work (e.g., it could be a function of T and g).

Figure 6 (left) visualizes the size of the 309 files that we inserted, and bars are colored according to the outcome of the test. The relay and uploader tests are shown in separate facets. Overall, we did not observe that manifest size influences the FPR or TPR directly. Figure 6 (right) demonstrates that the TPR is influenced significantly by the number of insert requests received by a neighbor.

5.3 Analytically Derived FPR and Power

We now derive the False Positive Rate and Power [46] of our hypothesis test, which are $O\left(Exp(\frac{-T}{g\ln(h)})\right)$ and $\Omega\left(1 - Exp(\frac{g^3}{T})\right)$, respectively. Recall that for Eq. 2, hypothesis $H1$ is that the subject is the requester, and hypothesis $H2$ is that the subject is a relayer. We label the subject as the requester if the number of requests observed $X = r$ is large enough such that $Pr(H1|X = r) > t$ for a threshold $0.5 < t \leq 1$. The probability of accepting $H1$ given that $H2$ is true, (i.e., a false positive or Type I error) is

$$Pr(\text{accept } H1|H2 \text{ is true}) = Pr(r \geq \varrho|H2 \text{ is true}),$$

where ϱ is the smallest number of requests that results in a positive test given: T total requests from the true requester; g neighbors of the subject; a hypothesis of h neighbors of the requester; and $\mathbf{h}$ actual neighbors (in bold) of the requester.

Power is the complement of the false negative probability:

$$1 - Pr(\text{accept } H2|H1 \text{ is true}) = 1 - Pr(r < \varrho|H1 \text{ is true}).$$

Therefore, for both statistics, we require an expression for ϱ.

THEOREM 2: *The probability of a false positive for Eq. 2 is the probability that a relayer issues at least ϱ requests, which is*

$$Pr[X \geq \varrho] = 1 - F\left(\lceil\varrho\rceil - 1; T, {}^{1}/_{gh}\right). \quad (4)$$

And the Power of Eq. 2 is the probability that an actual requester issues at least ϱ requests, which is

$$Pr[X \geq \varrho] = 1 - F\left(\lceil\varrho\rceil - 1; T, {}^{1}/_{g}\right). \quad (5)$$

Note that $F(k; n, p) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}$. We omit the proof because these statements follow directly from the binomial CDF given an expression for ϱ, which we now derive. For convenience, let $\tau = (1-t)/t$.

THEOREM 3:

$$\varrho = \frac{T\ln\left(\frac{gh-h}{gh-1}\right) + \ln(\tau/g)}{\ln\left(\frac{gh-h}{gh-1}\right) - \ln(h)} \quad (6)$$

The proof appears in Appendix B.

We characterize the asymptotic behavior of the two statistics. As we prove below, the FPR decreases exponentially with T, and Power approaches 1 from below exponentially with T.

In our proofs, we use two simpler formulations of ϱ, as stated in these lemmas.

> **LEMMA 1:** *The value of ϱ, the smallest number of requests that results in a positive test given T, g, and h, is bounded by*
>
> $$\varrho > \frac{T/(2g)}{{}^{1}\!/_{2g} + \ln(h)} \qquad (7)$$

The proof appears in Appendix B.

> **LEMMA 2:** *The value of ϱ, the smallest number of requests that results in a positive test given T, g, and h, is bounded by*
>
> $$\varrho < {}^{T}\!/_{g} - g \qquad (8)$$
>
> *for $g \geq 2, h \geq 2$, and $0 < \tau < 1$.*

The proof appears in Appendix B.

> **THEOREM 4:** *For Eq. 2, the FPR is $O\left(Exp\left(\frac{-T}{g\ln(h)}\right)\right)$.*

PROOF: We use the following Chernoff bound (proven as Theorem 6 in Appendix D).

$$Pr[X \geq \varrho] = Pr[X \geq (1+\delta)\mu] \leq Exp\left(-{}^{1}\!/_{3}\,\delta\mu\right), \text{ for } \delta \geq 1$$

In the case of a false positive, a relayer has $\mu = E[X] = {}^{T}\!/_{g\mathbf{h}}$. We apply this bound to Eq. 4 by selecting a value for δ such that having $X = (1+\delta)\mu$ or more observations is equivalent to a false positive. We do this by setting $(1+\delta)\mu = \varrho$ (via Lemma 1) and solving for δ:

$$\begin{aligned}(1+\delta)\frac{T}{g\mathbf{h}} &= \frac{T/(2g)}{{}^{1}\!/_{2g} + \ln(h)}\\ \delta &= \frac{g\mathbf{h}}{1 + 2g\ln(h)} - 1\end{aligned}$$

Note that ϱ could be larger per Lemma 1; larger values of δ would further reduce the probability of $X > (1+\delta)\mu$. Substituting this value for δ back into the bound, we see that

$$Pr[X \geq \varrho] \quad \leq \quad Exp\left(-\left(\frac{g\mathbf{h}}{1+2g\ln(h)} - 1\right)\frac{T}{3g\mathbf{h}}\right) \qquad (9)$$

which is bounded by the equation in the theorem's statement (when $\delta \geq 1$). Note that while this bound is always valid, it is not tight (i.e., it is greater than one) when $g\mathbf{h} \leq 1 + 2g\ln(h)$. □

We bound Power from below, as we wish to know the probability that the actual requester does not issue at least ϱ requests. We seek to characterize the asymptotic behavior of Power as T increases.

> **THEOREM 5:** *For Eq. 2, the Power is $\Omega\left(1 - Exp\left(\frac{g^3}{T}\right)\right)$ if $T > 0, 2 \leq g < \sqrt{T}, h \geq 2$, and $0 < \tau < 1$.*

The proof appears in Appendix C.

Using Eqs. 4 and 5, we plot in Figure 7 the exact FPR and Power for selected values of g, $\mathbf{h}$, T, and h. Dotted lines show Chernoff bounds from Eqs. 9 and 17 for $g = 8$. As predicted by Theorem 4, the semi-logscale plot illustrates that the FPR is exponentially decreasing with T and exponentially increasing with g. Increases in $\mathbf{h}$ reduce the error rate. An increase in h would increase the FPR in a less significant fashion. The relative value of $\mathbf{h}$ and h matter, but most importantly, the exact FPR values are predicted to be small in scenarios we observe in Freenet. In Freenet, T is linked to file size (each block is 32KB). Our analysis predicts a low FPR for scenarios that concern us on Freenet: images and videos. The equations (and plots) also predict high Power for almost any combination of h, T, and g, orders of magnitude higher than the FPR.

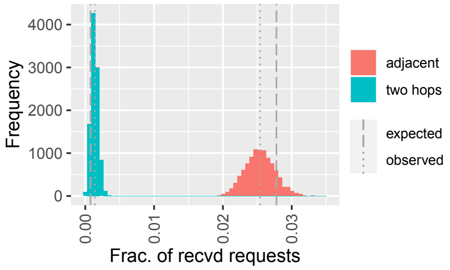


Figure 8: Distribution of requests received by observer either adjacent to or two hops away from the requester (degree 36). Visually, it's clear observers can differentiate the scenarios.

This analysis assumes that the parameters of the model are correct. Our FPR and TPR results from Sections 5.1 and 5.2 make no such assumption, and neither do our results by simulation in Section 6. All show results consistent with the theorems above.

6 EVALUATION II: SIMULATIONS

To gain further insight into the accuracy of our investigative method, we designed and executed a custom simulator. We constructed a Freenet-like topology and performed FOAF routing for thousands of concurrent downloaders. We measure the FPR and TPR. These evaluations agree with Section 5's results but lend different insights.

Compared to our in situ tests, our simulation has the advantage that we can modify a variety of parameters. To highlight this difference, we modified various parameters in the single downloader and concurrent downloader scenarios below, and we present the results differently.

Assumptions. Freenet is designed to create a small-world topology [47]. In our simulations, we create a small-world topology via Watts and Strogatz's algorithm [48]; numerous evaluations of Freenet simulate over small-world topologies [37, 38, 49–51]. Specifically, we assign each node a random location in the location space from [0, 1). We then assign each node a set of edges: c edges to *close* nodes and l edges to *long-distance* nodes, where distance is defined using the Freenet distance metric (Section 2). The c edges to close nodes are chosen uniformly at random from among the $2c$ closest nodes, and the l edges are chosen uniformly at random from among the remaining nodes. Each node thus has *at least* $c + l$ edges and typically more.

All of our graphs were constructed with 5,000 nodes total. In any single trial, all nodes in a graph had the parameters $c + l$ of either 10+1, 18+2, 27+3, 54+6, or 81+9, resulting in average degrees of 12, 24, 36, 72, and 108, respectively. For each degree, we constructed 250 random graphs. The real Freenet graph is comprised of nodes with a variety of degrees, at or below these values.

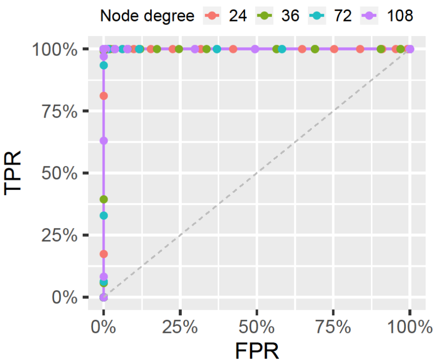


Figure 9: ROC curve for investigator success for Eq. 2, from trials of 500 small-world graphs per degree. (T = 1000.)

6.1 Single Downloader

For each graph, we requested blocks from a random node to one of the 5,000 locations in the graph. To find the content, we use the Freenet friend-of-a-friend routing algorithm, HTL decrementing, and other important details from the Freenet system (Section 2).

Specifically, we ran two types of trials, each type on 250 instances of each graph size. In the first type of trial, for a specific graph, we selected a node at random as an originator, another at random as a destination of the request, and then one of the originator's peers at random as an observer. In the second trial type, we selected a node at random as an originator, another at random as a destination, then one of the originator's peers *two hops* away at random as an observer. We ran 5,000 trials of each type for each graph instance. For each trial, we simulated 5,000 requests that resulted in a path that did or did not include the observer. From the 5,000 requests, we constructed 10,000 bootstrap samples for error analysis [52]. We constructed each bootstrap sample by selecting 5,000 requests uniformly at random with replacement from a trial's 5,000 requests.

Results. Figure 8 provides an intuitive explanation of why requesters are distinguishable from relayers. The figure shows the results for exactly two trials. The plot shows, for each trial, the fraction of requests observed from the 10,000 bootstrapped samples. One distribution of bootstrapped samples is presented for the case when an observer is adjacent to the requester, and another distribution for when it is two hops away. A simple visual test distinguishes the two scenarios. For these experiments, observations included only requests with HTLs of 18 or 17.

In Figure 8, vertical lines show the observed mean fraction of requests for the two scenarios, and two other lines show a simple expected fraction of 1/degree for the adjacent and $(1/\text{degree})^2$ for the two-hop case. Because nodes in the graph have different degrees, and requests are sent from a random node, there is not an exact match with the mean degree.

Figure 9 shows a Receiver Operating Characteristic (ROC) curve for the result of applying Eq. 2 to all trials for each size graph. The plot shows FPR versus TPR as a curve parameterized by a threshold from 0 to 1 for the value of Eq. 2. A true positive is an observation from an adjacent node with a probability greater than or equal to the threshold. A false positive is an observation from a two-hop neighbor with the same. As the figure shows, the algorithm obtains near-perfect accuracy in simulation. The area under the curve is numerically equivalent to 1.0 for all four graph types.

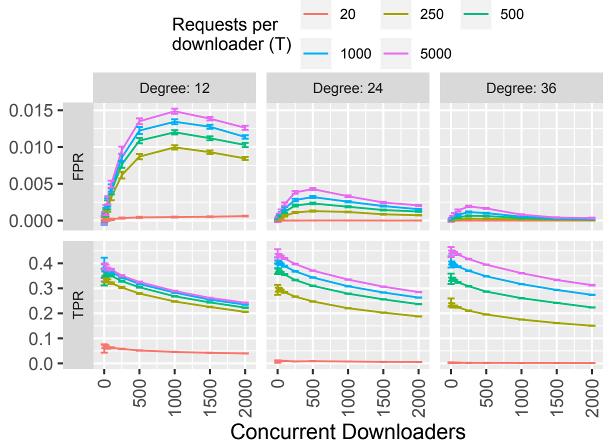


Figure 10: Simulations of concurrent downloaders of the same manifest (5000-node topology). The FPR remains below 1.5% for all scenarios, and decreases as the average degree of the topology increases. Error bars show 95% c.i.

6.2 Concurrent Downloaders

Figure 10 shows the TPR and FPR of our method in the presence of concurrent downloaders from simulations. All simulations are over 5000-node small-world topologies constructed using a variety of degrees. The simulations vary the number of requests per downloader T (as distinct lines on the plots) and the number of concurrent downloaders (as the independent variable on the x-axis). Our simulations take into account Eq. 3, which *(i)* counts the distinct blocks requested (and not the total number of requests), and *(ii)* discounts when duplicate requests for the same block are received. In no case does the FPR rise above 1.5%, and the FPR lowers as node degree increases. As the size of the manifest and T increase, the FPR increases slowly. If our simulation enforced request caching, the FPR would be lower[3]

We did not evaluate concurrent uploaders. We leave such an evaluation for future work based on the following details. Concurrent CHK uploaders would increase the possibility of a false positive, since each upload would use the same encryption key. But nodes assign unique identifiers (UIDs) to their inserts. If an observer received duplicate content blocks (Section 4.4) having distinct UIDs, she could assume concurrent uploaders and discard the run.

7 PROSECUTIONS USING THIS METHOD

Our method has been successfully used to investigate and prosecute cases. Convictions include: engaging in illicit sexual conduct in foreign places with minors (i.e., child sex tourism); production of CSAM with the intent to transport to the US; and possession, receipt, and distribution of CSAM. Table 2 summarizes five cases brought in US Federal District courts. Each involved a court-approved search warrant that was supported by the results of our method. We have omitted cases that are still in progress at the district court level. In four, the defendants filed motions to suppress the evidence obtained

[3] In our prior work [14], we derived a model of concurrent downloaders. Eq. 7 is corrected as $\sum_{i=1}^{n} \frac{1}{2^i(b+1)} = \frac{(1-2^{-n})}{b+1}$, which is indeed less than the expected fraction of requests from a true downloader, $1/(b+1)$. The comparison of expectations does not capture variance, but our simulations do and replace it.

Case	Evidence Discovered	Outcome	Notes
US v. Dickerman *US Court of Appeals Eighth Circuit, No. 18-3150; and Eastern Dist. of Missouri, Case 4:16-CR-258*	Over 600 CSAM images and videos.	Guilty plea. 60 months imprisonment, $5,000 restitution to victims.	Motion to suppress denied. Denial upheld by US Court of Appeals for the Eighth Circuit.
US v. Hall *Dist. of Maryland, Case 1:16-CR-469*	CSAM of children known to Hall on his camera. Over 1,000 additional CSAM images.	Guilty plea. 300 months imprisonment, $125,000 restitution.	Motion to suppress denied. Two children rescued.
US v. Popa *Northern Dist. of Ohio, Case 5:18-CR-448*	6,222 CSAM images and videos; 239,094 additional exploitative images (age difficult or child erotica).	Guilty plea. 150 months imprisonment.	Motion to suppress denied.
US v. Rogers *Northern Dist. of Ohio, Case 3:18-CR-26*	CSAM images and videos.	Found guilty by a jury. 96 months imprisonment, $80,000 restitution.	Search warrant uncontested. Admitted to downloading CSAM during the trial.
US v. Weyerman *Eastern Dist. of Pennsylvania, Case 2:19-CR-88*	CSAM images found on computers and external drives.	Guilty plea (reserving the right to appeal). Sentencing scheduled.	Motion to suppress denied. Defendant was on parole for child rape conviction.

Table 2: Freenet cases in US Federal courts where our method was applied and in which there was a hearing or trial. The evidence discovered column lists the results of executing a court-authorized search warrant.

from the search, a hearing was held, and the courts upheld the search warrants and our method explicitly. In a fifth case the search, and therefore our method, was not contested. We assisted with all five cases; in four we testified under oath, discussing our prior work [14]. In the earliest case, a Freenet contributor assisted the defense. We are not aware of any court that has ruled our method invalid or not forensically sound.

8 AVOIDING DETECTION

In this section, we discuss Freenet changes that might be proposed to prevent the detection of requesters. We detail why they may not be sufficient. It is our opinion that Freenet is likely not repairable without significant architectural changes.

Changing the Maximum HTL. To prevent a statistical attack on the HTL, Freenet randomizes decrementing the initial HTL (§2). We prove in Theorem 1 that this algorithm is effective. In this paper, we have focused on HTLs of 18 and 17, however these values are merely for clarity; our technique works for any HTL. Further, if a user changes their maximum HTL to something larger than the current default of 18, its neighbors will immediately reduce it to the default and these large values will be noticeable by investigators. Configuring a lower HTL is also ineffective because it is easy to observe the maximum value: HTLs that exceed the maximum are lowered before being forwarded.

Randomizing the HTL. To identify a node as a requester, we rely on consistent HTL values per manifest being referenced (§4.4). Freenet could choose to decrement HTLs on a per-packet basis by some integer $d \geq 0$, requiring the investigator to include requests with a range of HTLs in each run, possibly increasing the FPR. To minimize the FPR, this approach would force an investigator to use a threshold that drives down the TPR as well. But the approach is not a clear win. First, we expect the investigator could statistically infer the requester, given the HTLs for each request associated with a particular manifest. On average, higher HTLs would be expected from the originator. It's non-obvious what algorithm selects d in a way that prevents this inference. Including multiple HTLs in runs could increase the FPR if such an algorithm were implemented. However, our basic approach would still apply and would remain effective, especially for manifests that are less popular. Quantifying whether the FPR increase is significant would require an update of our analysis in Section 4 to include manifest popularity. We leave this analysis for future work.

Replacing HTLs with Probabilistic Forwarding. Freenet could be redesigned completely to remove HTLs from requests, like OneSwarm [53], which also attempted anonymous file sharing. Like Freenet, OneSwarm peers search for pieces of a desired file distributed among nodes. The requests are partially flooded: requests are forwarded along each neighboring edge according to a set probability. A random delay obscures whether a node replied based on their own stored content, or because they were acting as a relayer. OneSwarm shut down after research showed that it was subject to three distinct de-anonymization techniques [54, 55].

Using Darknet Mode. The Freenet developers encourage the use of darknet mode, where most, if not all neighbors of a node are fully trusted [56]. This approach may help avoid detection by strangers, but it is hardly a realistic approach to maintaining a viable, fully connected network. Regardless, our algorithm is not specific to opennet, and could be employed by a neighboring darknet peer.

Removing FOAF Routing. If FOAF information were not exchanged, g would not be observable. A two-peer attack, where each peer manufactures requests to be relayed by the subject, could be used to estimate the fraction of T expected by each peer. From there, a version of our method could be employed successfully.

Masking IP Addresses. A requester could hide their IP address behind an anonymizing service, such as a VPN. This strategy would cause our method to associate the download with a VPN address instead of a home address. VPNs reduce the investigator's task to a single point. Further, VPNs do not always work as advertised [57], and the amount of security provided is challenging for consumers to evaluate.

9 RELATED WORK

While prior work has investigated vulnerabilities in Freenet, our work is unique in several ways. In short, we present the only method based on traffic passively received at a single observer with a justified and proven approach.

Borisov [58] applied information theoretic metrics [59, 60] to Freenet and found it to have relatively low anonymity. McCoy [40] investigated de-anonymizing Freenet. In our notation, McCoy declares a subject to be the actual requester if $o\frac{\ln(T)}{T} > 3.3$ during a fixed duration of time without regard to HTL. This ad hoc approach took advantage of a bug present at the time in an old version of the request routing algorithm. The routing algorithm has been entirely replaced to use locations and the FOAF system. Regardless, it could not be used as a forensically sound approach.

Tian et al. [36–38, 49] discovered a Traceback Attack in Freenet that exploits a unique identifier (UID) assigned to each request, normally used to detect routing loops, as confirmation that the peer must have been on the path from the original requester. The approach requires actively probing all neighbors of the peer, and leveraging a Routing Table Insertion (RTI) attack in which the attacker traverses the network toward the requester. RTI is an approach due to Baumeister et al. [35, 51] that does not de-anonymize peers. The RTI vulnerability can be addressed with randomized routing [61]. Freenet developers addressed the Traceback attack by having peers discard a UID after receiving the response to the outstanding request. Discarding UIDs does not address our method.

Roos et al. [39] show how Freenet network probes, intended to gather obfuscated node values, can be used to infer the actual value with a Bayesian model after multiple observations. The attack is a general approach, but they provide a specific example of using the approach to infer the bandwidth of peers, which could be used to detect opennet-darknet bridges. Roos et al. [43] offers statistics on Freenet session length and churn.

OneSwarm [53] is an anonymous file sharing network with similarities to Freenet. It provides anonymity by removing HTL counts, obfuscating timing delays, and flooding probabilistically; we have shown that OneSwarm's approach is not secure [54, 55].

10 ETHICAL CONCERNS

This work raises a variety of ethical concerns. We consulted and adhered to strict ethical standards along many dimensions.

Human Subjects Research. We followed the policies and guidance of our IRB, who explicitly approved our research.

Legal Standards of Forensic Soundness. *Daubert* [34] is the primary standard followed by US Courts when evaluating the scientific conclusions of an investigator. We have adopted this standard as our definition of forensic soundness by: *(i)* publishing our method earlier in a peer-reviewed IEEE workshop [14] and submitting this extended version for peer review; *(ii)* basing our methods on a testable hypothesis; *(iii)* stating a known error rate for our method; *(iv)* following an existing set of standards for our method; *(v)* and using methods generally accepted within the scientific community. There are many further considerations that are relevant to criminal prosecutions and often case specific. While we cannot list them all here, as noted in Section 7, cases that were initiated based on our method have been upheld by US Federal Courts despite motions to the contrary and lengthy consideration.

Mitigating Harms. For many years, Freenet has been prominently used as a forum for the sexual exploitation of children. Our providing support to law enforcement follows universal ethics that children should be rescued from harm. It also is aligned with the ACM Code of Ethics [62], which instructs members to "ensure all harm is minimized" and have an "obligation to report any signs of system risks that might result in harm."

Responsible Disclosure. Our method has been disclosed publicly [14] and Freenet developers are aware of it. To our knowledge, they have taken no actions because of it. Since before our disclosure, Freenet's website has warned users that it provides limited protection and is vulnerable [63] in language so broad that it covers our work. The installed software also warns that "it may be quite easy for others to discover your identity" and that it is connecting to "strangers" and lists neighbors as "untrusted peers" [41].

Ethics of the Freenet Project. Freenet offers virtually no protection against the weakest of adversaries who would misuse our method. As such, Freenet should not be used by vulnerable persons requiring censorship resistance and anonymity for ethical purposes, such as dissidents. At the same time, Freenet is used rampantly by those who sexually exploit children. From either viewpoint, the societal value of Freenet is negligible. The Freenet developers ask on their own web pages, "What about child porn?" They answer, "How people choose to use the tool is their sole responsibility. As a communication medium, Freenet cannot be considered responsible for what people use it for." [63] We disagree. The developers have responsibilities, especially when, perennially, the use consists of vast, unmitigated harms to society's most vulnerable persons. A similar analysis of harms, protections, and responsibility can be applied to the Tor Project and Onion Services in particular; see [20, 22].

11 CONCLUSIONS

We designed and evaluated a method for forensic investigation of CSAM-based privacy violations and re-victimization on Freenet. We observed that about one-third of the traffic on Freenet consists of CSAM-related requests. Our method is distinguished from past work in Freenet in that it can identify both uploaders and downloaders, by its formal basis, and by its minimal resource requirements: a single, passive peer that uses only the traffic sent to it. Our evaluation is motivated by forensic soundness and is extensive. It includes: in situ measurement of both real CSAM downloads and our own set of (non-CSAM) uploads; derivation of the FPR and Power of our hypothesis test; simulations of single and concurrent downloaders; and characterization of the network to inform parameter selection. We observed an FPR of 0.002 ± 0.003 for identifying downloaders. For identifying uploaders, we observed an FPR of 0.009 ± 0.018, a precision of 0.9998 ± 0.01, and a TPR of 0.44 ± 0.01. We were consulted during investigations using our method, and we were participants in several US Federal Court cases, which brought a modicum of justice to many child victims of Freenet users.

This material is based upon work supported in part by the National Science Foundation under Award No. 1816851 and in part by a Signature Interdisciplinary Research Area Grant from the Rochester Institute of Technology.

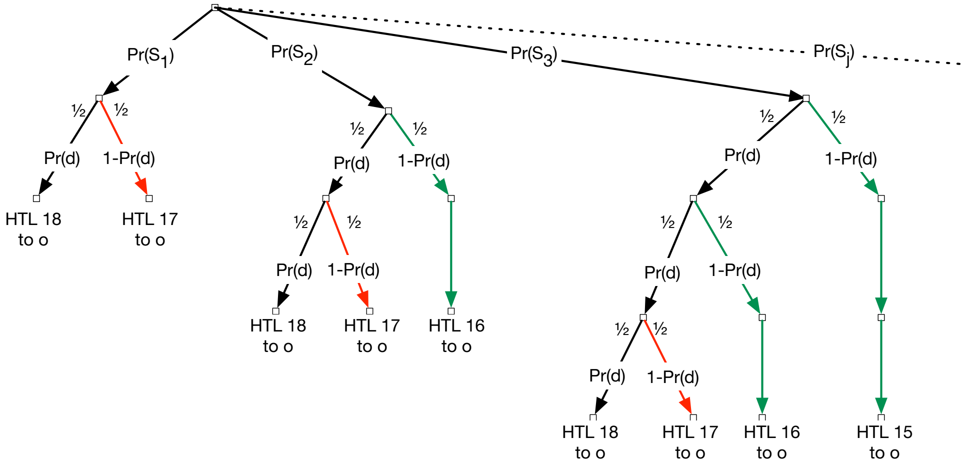


Figure 11: Sample space of the HTL of requests received at a peer o. Red lines indicate that HTLs are decremented.

APPENDIX

A ANALYSIS OF HTLS OF 18 AND 17

Consider a topology with a path of adjacent peers such that observer o is connected to subject S_1, who is connected to S_2, who is connected to S_3, and so on for j peers. In this topology, requests received by o may have come directly from S_1, or relayed by S_1 for S_2, and so on: $o \leftarrow S_1 \leftarrow S_2 \leftarrow \ldots \leftarrow S_j$. The HTL of requests received at o depends on who is the requester and whether the edges between peers decrements requests with HTLs of 18. This sample space is illustrated in Figure 11, where $Pr(d)$ is the probability of decrementing an 18, and $Pr(S_i)$ is the probability that a node at level j is the original downloader or uploader.

THEOREM 1: *The probability that hypothesis $H1$ is true is no different when a run is composed of requests with HTLs of 18 or 17, regardless of the distance of the requester from the observer, if the probability is ${}^1\!/_2$ that an 18 is decremented; i.e.,*

$$Pr(H1 \text{ is true} \,|Y = 17) = Pr(H1 \text{ is true} \,|Y = 18).$$

PROOF: Let $Pr(d) = 1/2$ represent the probability that a given peer does not decrement HTLs of 18 on a particular edge. Note that $Pr(Y = 18|H1)$= $Pr(d) = {}^1\!/_2$, and similarly that $Pr(Y = 17|H1)$ =$1-Pr(d) = {}^1\!/_2$. From Bayes' rule we know that

$$Pr(H1|Y = 18) \quad = \quad \frac{Pr(Y = 18|H1)Pr(H1)}{Pr(Y = 18)} = \frac{{}^1\!/_2 Pr(H1)}{Pr(Y = 18)}.$$

Similarly, we have

$$Pr(H1|Y = 17) \quad = \quad \frac{Pr(Y = 17|H1)Pr(H1)}{Pr(Y = 17)} = \frac{{}^1\!/_2 Pr(H1)}{Pr(Y = 17)}.$$

Therefore, we need only show that $Pr(Y = 18) = Pr(Y = 17)$. Let $Pr(S_j)$ represent the probability that peer S_j is the downloader. Following the black arrows in Figure 11 we have for Pr(Y=18)

$$\begin{aligned} Pr(Y = 18) \quad &= \quad Pr(S_1)Pr(d) + Pr(S_2)Pr(d)Pr(d) + \ldots + \\ &\quad\quad Pr(S_j)Pr(d)^j \\ &= \quad \sum_{i=1}^{j} Pr(S_i)({}^1\!/_2)^i. \end{aligned} \tag{10}$$

For all HTL 17s received we have

$$\begin{aligned} Pr(Y = 17) \quad &= \quad Pr(S_1)(1-Pr(d)) + Pr(S_2)Pr(d)(1-Pr(d)) + \ldots + \\ &\quad\quad Pr(S_j)Pr(d)^{j-1}(1 - Pr(d)) \\ &= \quad \sum_{i=1}^{j} Pr(S_i)({}^1\!/_2)^{i-1}({}^1\!/_2) \\ &= \quad \sum_{i=1}^{j} Pr(S_i)({}^1\!/_2)^i. \end{aligned} \tag{11}$$

Thus, Eqs. 10 and 11 are equal. Further, this result holds when $Pr(d) = 1/2$ only. □

B ϱ AND ITS BOUNDS

THEOREM 3:

$$\varrho = \qquad \frac{T \ln\left(\frac{gh-h}{gh-1}\right) + \ln(\tau/g)}{\ln\left(\frac{gh-h}{gh-1}\right) - \ln(h)} \tag{12}$$

PROOF: We derive an expression for ϱ from Eq. 2, starting from threshold $t = Pr(H1|X = \varrho)$.

$$t = \frac{Pr(H1)Pr(\varrho|H1)}{Pr(H1)Pr(\varrho|H1) + Pr(H2)Pr(\varrho|H2)}$$

$$Pr(H2)Pr(\varrho|H2) = \frac{1-t}{t} Pr(H1)Pr(\varrho|H1)$$

$$\frac{g}{g+1}\text{Binom}(\varrho; T, {}^1\!/_{gh}) = \frac{1-t}{t}\,\frac{1}{g+1}\text{Binom}(\varrho; T, {}^1\!/_g)$$

For clarity, let $\tau = (1 - t)/t$. We know that $g > 1$ by observation and we expect $h > 1$ (see §4.4). Using this fact, we get:

$$\binom{T}{\varrho}\left(\frac{1}{gh}\right)^{\varrho}\left(1 - \frac{1}{gh}\right)^{T-\varrho} \quad = \quad \frac{\tau}{g}\binom{T}{\varrho}\left(\frac{1}{g}\right)^{\varrho}\left(1 - \frac{1}{g}\right)^{T-\varrho}$$

from which we can solve for ϱ to yield Eq. 12. □

Below, we provide a lower bound on ϱ. We use Lemma 1 in Section 5.3.

LEMMA 1: *The value of ϱ, the smallest number of requests that results in a positive test given T, g, and h, is bounded by*

$$\varrho > \frac{T/(2g)}{1/2g + \ln(h)} \tag{13}$$

PROOF: We begin by using the well-known inequality $\ln(1+x) < x$ for all $x > -1$, and we compute:

$$\begin{aligned} \ln(1+x) &< x \\ \ln\left(\frac{gh-h}{gh-1}\right) &< \frac{gh-h}{gh-1} - 1 \\ &< \frac{1-h}{gh-1} \\ &< -\frac{1}{2g} \text{ for } h \geq 2 \end{aligned}$$

For simplicity, we define $X = -\ln\left(\frac{gh-h}{gh-1}\right)$. Then we see that $X > \frac{1}{2g}$ for all $h \geq 2$. Substituting into Eqn. 12 from Section 5.3, and solving for X:

$$\begin{aligned} \varrho &= \frac{T\ln\left(\frac{gh-h}{gh-1}\right) + \ln(\tau/g)}{\ln\left(\frac{gh-h}{gh-1}\right) - \ln(h)} \\ &= \frac{-TX + \ln(\tau/g)}{-X - \ln(h)} \\ X &= \frac{\varrho\ln(h) + \ln(\tau/g)}{T - \varrho} \end{aligned}$$

We then have:

$$\begin{aligned} \frac{\varrho\ln(h) + \ln(\tau/g)}{T-\varrho} &> \frac{1}{2g} \text{ for } h \geq 2 \\ 2g\varrho\ln(h) + 2g\ln(\tau/g) &> T - \varrho \text{ given } g > 0, T > \varrho \\ \varrho &> \frac{T/2g - \ln(\tau/g)}{1/2g + \ln(h)} \\ \varrho &> \frac{T/2g}{1/2g + \ln(h)} \text{ given } \frac{\tau}{g} < 1 \end{aligned}$$

□

Below we provide an upper bound on ρ. We use Lemma 2 in Section 5.3.

LEMMA 2: *The value of ϱ, the smallest number of requests that results in a positive test given T, g, and h, is bounded by*

$$\varrho < T/g - g \tag{14}$$

for $g \geq 2, h \geq 2$, and $0 < \tau < 1$.

PROOF:

$$\frac{T}{g} - g > \varrho$$

$$\frac{T}{g} - g > \frac{T\ln\left(\frac{gh-h}{gh-1}\right) + \ln(\tau/g)}{\ln\left(\frac{gh-h}{gh-1}\right) - \ln(h)}$$

Let $a = \ln\left(\frac{gh-h}{gh-1}\right)$, $b = \ln(\tau/g)$, and $c = a - \ln(h)$. Since $g > 1, h > 1$, and $0 < \tau < 1$, we know that $a < 0, b < 0$, and $c < 0$.

$$\frac{T}{g} - g > \frac{Ta+b}{c}$$
$$T(c - ga) < g^2c + gb$$

We prove below that $c - ga < 0$, and so we flip the inequality in the following.

$$T > \frac{g^2c + gb}{c - ga} \tag{15}$$

For any values of $g > 1, h > 1$, and $0 < \tau < 1$, there exists a T that satisfies Inequality 15 since the RHS is a constant.

To complete the proof, we now show that $c - ga < 0$.

$$c - ga < 0$$
$$a - \ln(h) - ga < 0$$
$$(1-g)\ln\left(\frac{gh-h}{gh-1}\right) - \ln(h) < 0$$

To prove this inequality, we let $f(g,h) = (1-g)\ln\left(\frac{gh-h}{gh-1}\right) - \ln(h)$. Note that $f(2,2) \approx -0.29 < 0$. From this initial point, we show that larger values of g and h are also negative.

The partial derivative w.r.t. h is

$$\frac{\partial f}{\partial h} = \frac{g - gh}{h(gh-1)}.$$

It is always negative, and therefore the value of $f()$ becomes more negative as h increases.

The argument w.r.t. g is more complicated. As stated above, we know that $f(2,2) \approx -0.29$. Note that

$$\lim_{g\to\inf} f(g,h) = 1 - 1/h - \ln(h),$$

which is less than zero when $h \geq 2$. For example, $1 - 1/2 - ln(2) \approx -0.19$. We must show that $f()$ is always negative for $g > 2$; in other words, we must show that $\frac{\partial f}{\partial g}$ is always positive.

The partial derivative w.r.t. g of $f()$ is

$$\frac{\partial f}{\partial g} = -\frac{h-1}{gh-1} - \ln\left(\frac{(g-1)h}{gh-1}\right).$$

We see that $\frac{\partial f}{\partial g}$ starts out positive. Specifically, $\frac{\partial f(2,2)}{\partial g} \approx 0.072$. Note that $\lim_{g\to\inf} \frac{\partial f}{\partial g} = 0$. We must show that $\frac{\partial^2 f}{\partial g}$ is always negative.

$$\frac{\partial^2 f}{\partial g} = -\frac{(h-1)^2}{(g-1)(gh-1)^2}$$

Thus, $\frac{\partial^2 f}{\partial g}$ is always negative. In sum, $\frac{\partial f(2,2)}{\partial g}$ is positive, and $\frac{\partial f}{\partial g}$ approaches 0 in the limit that g approaches infinity, and it is monotonically decreasing. Since $\frac{\partial f}{\partial g}$ is always positive, we know that starting from $f(2,2)$, which is a negative value, $f()$ approaches a negative value in the limit that g approaches infinity and is monotonically increasing; that is, $f()$ is never positive. □

C BOUND OF POWER

THEOREM 5: *For Eq. 2, the Power is* $\Omega\big(1 - Exp\left(\frac{g^3}{T}\right)\big)$ *if* $T > 0, 2 \le g < \sqrt{T}, h \ge 2,$ *and* $0 < \tau < 1$.

PROOF: We bound the Power probability from below. That is, we show that the probability of an actual requester issuing less than ϱ falls exponentially. We make use of a well-known Chernoff bound [64].

$$Pr[X \le \varrho] \quad = \quad Pr[X \le (1-\delta)\mu] \le Exp(\delta^2\mu/2), \qquad (16)$$

for $0 < \delta < 1$. We let $\mu = {}^{T}/_{g}$, and using Lemma 2, we solve for δ.

$$\begin{aligned}(1-\delta)\mu &= {}^{T}/_{g} - g \\ \delta &= \frac{g^2}{T}\end{aligned}$$

We apply the bound [64].

$$\begin{aligned}Pr[X \le \varrho] = Pr[X \le (1-\delta)\mu] &\le Exp\left(\frac{\mu}{2}\delta^2\right) \\ &= Exp\left(\frac{g^3}{2T}\right)\end{aligned}$$

Substituting, we have the following.

$$\begin{aligned}\text{Power} &= 1 - Pr[X \le \varrho] \\ &\ge 1 - Exp\left(\frac{g^3}{2T}\right) \qquad (17)\end{aligned}$$

Because $0 < \delta < 1$, the bound applies when $T > 0$ and $g < \sqrt{T}$. □

D CHERNOFF BOUND

The following proof of a Chernoff bound variant is due to Prof. Nick Harvey (https://www.cs.ubc.ca/~nickhar/W15/Lecture3Notes.pdf). We include it here for completeness since it is not in standard textbooks.

THEOREM 6:

$$Pr[X \ge (1+\delta)\mu] \le Exp\left(-{}^{1}/_{3}\,\delta\mu\right), \text{ for } \delta \ge 1.$$

PROOF: First, we claim that for $f(x) = (1+x)\ln(1+x) - x$, when $x \ge 1$ that $f(x) \ge x/3$. Consider the point $x = 1$. In that case, $f(1) = 2\ln(2) - 1 > 0.38 > 1/3$. Since the derivative $f'(x) = \ln(1+x)$ is greater than $\ln(2)$ for all $x \ge 1$, our claim is true. Next, we start with the well-known Chernoff bound [64]:

$$Pr[A \ge (1+\delta)\mu] \le \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}, \text{ for } \delta > 0.$$

We need only show that

$$\begin{aligned}\left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} &\le Exp(-{}^{1}/_{3}\,\delta\mu) \\ \ln\left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right) &\le \ln(Exp(-{}^{1}/_{3}\,\delta)) \\ (1+\delta)\ln(1+\delta) - \delta &\ge {}^{\delta}/_{3}\end{aligned}$$

The last inequality is true because of our earlier claim. □

REFERENCES

[1] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, “Freenet: A distributed anonymous information storage and retrieval system,” in *Proc. Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability (10.1007/3-540-44702-4_4)*, 2001, pp. 46–66.

[2] I. Clarke, S. Miller, T. Hong, O. Sandberg, and B. Wiley, “Protecting free expression online with Freenet,” *IEEE Internet Computing (10.1109/4236.978368)*, vol. 6, no. 1, pp. 40–49, Jan 2002.

[3] R. Dingledine, N. Mathewson, and P. Syverson, “Tor: The second-generation onion router,” in *Proc. USENIX Security Symposium*, 2004.

[4] S. Aked, “An investigation into darknets and the content available via anonymous peer-to-peer fifile sharing,” in *Proc. 9th Australian Information Security Management Conference (10.4225/75/57b52857cd8b3)*, December 2011.

[5] C. Guitton, “A review of the available content on tor hidden services: The case against further development,” *Computers in Human Behavior (10.1016/j.chb.2013.07.031)*, vol. 29, no. 6, pp. 2805–2815, 2013.

[6] A. Biryukov, I. Pustogarov, F. Thill, and R. Weinmann, “Content and Popularity Analysis of Tor Hidden Services,” in *Proc. IEEE Intl. Conference on Distributed Computing Systems Workshops (10.1109/ICDCSW.2014.20)*, 2014, pp. 188–193.

[7] M. Spitters, S. Verbruggen, and M. v. Staalduinen, “Towards a comprehensive insight into the thematic organization of the tor hidden services,” in *Proc. IEEE Joint Intelligence and Security Informatics Conference (10.1109/JISIC.2014.40)*, 2014, pp. 220–223.

[8] G. H. Owenson and N. J. Savage, “The Tor Dark Net,” Centre for International Governance Innovation https://www.cigionline.org/publications/tor-dark-net, Tech. Rep., September 30 2015.

[9] US Attorney’s Office, District of Nebraska, “2015 Annual Report, Project Safe Childhood,” https://www.justice.gov/usao-ne/file/830846/download, 2015.

[10] D. Moore and T. Rid, “Cryptopolitik and the darknet,” *Survival (10.1080/00396338.2016.1142085)*, vol. 58, no. 1, pp. 7–38, 2016.

[11] G. Owen and N. Savage, “Empirical analysis of tor hidden services,” *IET Information Security (10.1049/iet-ifs.2015.0121)*, vol. 10, no. 3, pp. 113–118, 2016.

[12] M. Bernaschi, A. Celestini, S. Guarino, and F. Lombardi, “Exploring and Analyzing the Tor Hidden Services Graph,” *ACM Trans. Web (10.1145/3008662)*, vol. 11, no. 4, Jul. 2017.

[13] Federal Bureau of Investigation, “‘Playpen’ Creator Sentenced to 30 Years,” https://www.fbi.gov/news/stories/playpen-creator-sentenced-to-30-years, May 5 2017.

[14] B. N. Levine, M. Liberatore, B. Lynn, and M. Wright, “Statistical detection of downloaders in freenet,” in *Proc. IEEE International Workshop on Privacy Engineering http://ceur-ws.org/Vol-1873/*, May 2017, pp. 25–32.

[15] J. Dalins, C. Wilson, and M. Carman, “Criminal motivation on the dark web: A categorisation model for law enforcement,” *Digital Investigation (https://doi.org/10.1016/j.diin.2017.12.003)*, vol. 24, pp. 62–71, 2018.

[16] G. Owenson, S. Cortes, and A. Lewman, “The darknet’s smaller than we thought: The life cycle of tor hidden services,” *Digital Investigation (10.1016/j.diin.2018.09.005)*, vol. 27, pp. 17–22, 2018.

[17] E. Bursztein, T. Bright, M. DeLaune, D. M. Eliff, N. Hsu, L. Olson, J. Shehan, M. Thakur, and K. Thomas, “Rethinking the detection of child sexual abuse imagery on the internet,” in *Proc. The World Wide Web Conference (10.1145/3308558.3313482)*, 2019, pp. 2601–2607.

[18] M. Faizan and R. A. Khan, “Exploring and analyzing the dark web: A new alchemy,” *First Monday (10.5210/fm.v24i5.9473)*, vol. 24, no. 5, Apr. 2019.

[19] S. He, Y. He, and M. Li, “Classification of Illegal Activities on the Dark Web,” in *Proc. International Conference on Information Science and Systems (10.1145/3322645.3322691)*, 2019, pp. 73–78.

[20] B. N. Levine, “Shining Light on Internet-based Crimes Against Children,” in *Proc. USENIX Security Symposium*, August 2019. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/levine

[21] B. R. da Cunha, P. MacCarron, J. F. Passold, L. W. dos Santos, K. A. Oliveira, and J. P. Gleeson, “Assessing police topological efficiency in a major sting operation on the dark web,” *Scientific Reports (10.1038/s41598-019-56704-4)*, vol. 10, no. 1, p. 73, 2020.

[22] B. N. Levine and B. Lynn, “Tor hidden services are a failed technology, harming children, dissidents and journalists,” in *Lawfare*, January 17 2020. [Online]. Available: https://www.lawfareblog.com/tor-hidden-services-are-failed-technology-harming-children-dissidents-and-journalists

[23] C. M. Steel, E. Newman, S. O’Rourke, and E. Quayle, “An integrative review of historical technology and countermeasure usage trends in online child sexual exploitation material offenders,” *Forensic Science International: Digital Investigation (10.1016/j.fsidi.2020.300971)*, vol. 33, 2020.

[24] U.S. Dept. of Justice, “The National Strategy for Child Exploitation Prevention and Interdiction: A Report to Congress,” http://www.projectsafechildhood.gov/docs/natstrategyreport.pdf pages 19–22, August 2010.

[25] ——, “The National Strategy for Child Exploitation Prevention and Interdiction: A Report to Congress,” https://www.justice.gov/psc/file/842411/download, April 2016.

[26] European Union Agency for Law Enforcement Cooperation, "Internet Organised Crime Threat Assessment," Europol, https://op.europa.eu/en/publication-detail/-/publication/d7582d31-1b04-11e9-8d04-01aa75ed71a1/language-en/format-PDF/source-88547505, Tech. Rep. (10.2813/858843), 2018.

[27] G. Bissias, B. N. Levine, M. Liberatore, B. Lynn, J. Moore, H. Wallach, and J. Wolak, "Characterization of Contact Offenders and Child Exploitation Material Trafficking on Five Peer-to-Peer Networks," *Child Abuse & Neglect (10.1016/j.chiabu.2015.10.022)*, vol. 52:185–199, Feb 2016.

[28] M. Cutajar, P. Mullen, J. Ogloff, S. Thomas, D. Wells, and J. Spataro, "Psychopathology in a large cohort of sexually abused children followed up to 43 years," *Child Abuse & Neglect (10.1016/j.chiabu.2010.04.004)*, vol. 34(11):813–822, 2010.

[29] G. Pérez-Fuentes, M. Olfson, L. Villegas, C. Morcillo, S. Wang, and C. Blanco, "Prevalence and correlates of child sexual abuse: a national study," *Comprehensive Psychiatry (10.1016/j.comppsych.2012.05.010)*, vol. 54, no. 1, pp. 16–27, 2013.

[30] E. Bazelon, "The price of a stolen childhood," *New York Times Magazine*, vol. https://nyti.ms/2kmwJlJ, Jan 27 2013.

[31] Phoenix 11, "Advocacy impact statement," https://protectchildren.ca/pdfs/C3P_Phoenix11_AdvocacyStatement_en.pdf.

[32] M. H. Keller and G. J. Dance, "'If Those Were Pictures of You, You Would Understand'," *New York Times*, Nov 9 2019, https://www.nytimes.com/2019/11/09/us/online-child-abuse.html.

[33] S. Young, "The use of normalization as a strategy in the sexual exploitation of children by adult offenders," *Canadian Journal of Human Sexuality*, vol. 6, 1997.

[34] *Daubert v. Merrell Dow Pharmaceuticals, Inc.* 509 U. S. 579, (1993).

[35] T. Baumeister, Y. Dong, Z. Duan, and G. Tian, "A Routing Table Insertion (RTI) Attack on Freenet," in *Proc. International Conference on Cyber Security (10.1109/CyberSecurity.2012.8)*, Dec 2012, pp. 8–15.

[36] G. Tian, Z. Duan, T. Baumeister, and Y. Dong, "A traceback attack on Freenet," in *Proc. IEEE INFOCOM (10.1109/INFCOM.2013.6566978)*, Apr 2013, pp. 1797–1805.

[37] ——, "Thwarting traceback attack on freenet," in *Proc. IEEE GLOBECOM (10.1109/GLOCOM.2013.6831161)*, Dec 2013, pp. 741–746.

[38] ——, "A traceback attack on Freenet," *IEEE Trans. on Dependable and Secure Computing*, vol. 14, no. 3, pp. 294–307, Jul 2017, 10.1109/TDSC.2015.2453983.

[39] S. Roos, F. Platzer, J. Heller, and T. Strufe, "Inferring obfuscated values in Freenet," in *Proc. International Conference and Workshops on Networked Systems (NetSys) (10.1109/NetSys.2015.7089062)*, Mar 2015, pp. 1–8.

[40] D. McCoy, "Anonymity analysis of freenet," UMI Number: 1439427, University of Colorado at Bolder, https://search.proquest.com/docview/305341725, 2006.

[41] Freenet reference daemon source code, https://github.com/freenet/fred.

[42] H. Zhang, A. Goel, and R. Govindan, "Using the small-world model to improve freenet performance," in *Proc. IEEE INFOCOM (10.1109/INFCOM.2002.1019373)*, 2002, pp. 1228–1237.

[43] S. Roos, B. Schiller, S. Hacker, and T. Strufe, "Measuring freenet in the wild: Censorship-resilience under observation," *Proc. Privacy Enhancing Technology Symposium (10.1007/978-3-319-08506-7_14)*, vol. LNCS 8555, pp. 263–282, Jul 2014.

[44] S. Dougherty, "Freenet statistics," https://www.asksteved.com/stats/.

[45] Tor metrics, https://metrics.torproject.org/userstats-relay-country.html.

[46] G. Casella and R. L. Berger, *Statistical inference.* Pacific Grove, CA: Brooks Cole, 2002.

[47] I. Clarke, O. Sandberg, M. Tosel, and V. Verendel, "Private communication through a network of trusted connections: The dark freenet," https://freenetproject.org/assets/papers/freenet-0.7.5-paper.pdf, Tech. Rep., 2010.

[48] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," *Nature (10.1038/30918)*, vol. 393(6684):440–442, 1998.

[49] G. Tian, Z. Duan, T. Baumeister, and Y. Dong, "Reroute on loop in anonymous peer-to-peer content sharing networks," in *Proc. IEEE Conf. Communications and Network Security (10.1109/CNS.2014.6997510)*, Oct 2014, pp. 409–417.

[50] O. Sandberg, "Distributed routing in small-world networks," in *Proceedings of the Meeting on Algorithm Engineering & Expermiments.* USA: Society for Industrial and Applied Mathematics, 2006, pp. 144–155.

[51] T. A. Baumeister, "Fundamental design issues in anonymous peer-to-peer distributed hash table protocols," Ph.D. dissertation, University of Hawai'i at Manoa, http://hdl.handle.net/10125/63489, 2019.

[52] B. Efron, "Bootstrap methods: another look at the jackknife," in *Breakthroughs in Statistics (10.1007/978-1-4612-4380-9_41).* Springer, 1992, pp. 569–593.

[53] T. Isdal, M. Piatek, A. Krishnamurthy, and T. Anderson, "Privacy-preserving P2P data sharing with OneSwarm," in *Proc. ACM SIGCOMM (10.1145/1851182.1851198)*, Aug 2010, pp. 111–122.

[54] G. Bissias, B. N. Levine, M. Liberatore, and S. Prusty, "Forensic Identification of Anonymous Sources in OneSwarm," *IEEE Trans. on Dependable and Secure Computing (10.1109/TDSC.2015.2497706)*, vol. 14, no. 6, pp. 620–632, Nov.-Dec. 2017.

[55] S. Prusty, B. N. Levine, and M. Liberatore, "Forensic Investigation of the OneSwarm Anonymous Filesharing System," in *Proc. ACM conference on Computer and communications security (CCS) (10.1145/2046707.2046731)*, Oct 2011, pp. 201–214.

[56] Freenet Project, https://wiki.freenetproject.org/, Feb 2017.

[57] M. Ikram, N. Vallina-Rodriguez, S. Seneviratne, M. A. Kaafar, and V. Paxson, "An Analysis of the Privacy and Security Risks of Android VPN Permission-enabled Apps," in *Proc. ACM Internet Measurement Conference (10.1145/2987443.2987471)*, 2016, pp. 349–364.

[58] N. Borisov, "Anonymous routing in structured peer-to-peer overlays," Ph.D. dissertation, Univ. of California, Berkeley, Berkeley, CA, Spring 2005. [Online]. Available: https://search.proquest.com/openview/bcb130965d4683ada51aa2aec50421a9

[59] C. Díaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proc. of Privacy Enhancing Technologies (10.1007/3-540-36467-6_5)*, 2002, pp. 54–68.

[60] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *Proc. of Privacy Enhancing Technologies (10.1007/3-540-36467-6_4)*, 2002, pp. 41–53.

[61] T. Baumeister, Y. Dong, G. Tian, and Z. Duan, "Using randomized routing to counter routing table insertion attack on freenet," in *Proc. IEEE GLOBECOM (10.1109/GLOCOM.2013.6831163)*, Dec 2013, pp. 754–759.

[62] Association for Computing Machinery, "ACM Code of Ethics and Professional Conduct," https://ethics.acm.org/, Adopted June 22 2018.

[63] Freenet Project, https://freenetproject.org/pages/help.html, Sep 2017.

[64] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*, 2nd, Ed. Cambridge University Press, 2017.