BITSTALKER: ACCURATELY AND EFFICIENTLY MONITORING BITTORRENT TRAFFIC

Kevin Bauer, Damon McCoy, Dirk Grunwald, and Douglas Sicker

University of Colorado, Boulder, CO, USA
{bauerk, mccoyd, grunwald, sicker}@colorado.edu

ABSTRACT

BitTorrent is currently the most popular peer-to-peer network for file sharing. However, experience has shown that BitTorrent is often used to distribute copyright protected movie and music files illegally. Consequently, copyright enforcement agencies currently monitor BitTorrent swarms to identify users participating in the illegal distribution of copyright-protected files. These investigations rely on passive methods that are prone to a variety of errors, particularly false positive identification.

To mitigate the potential for false positive peer identification, we investigate the feasibility of using *active* methods to monitor extremely large BitTorrent swarms. We develop an active probing framework called *BitStalker* that identifies active peers and collects concrete forensic evidence that they were involved in sharing a particular file. We evaluate the effectiveness of this approach through a measurement study with real, large torrents consisting of over 186,000 peers. We find that the current investigative methods produce at least 11% false positives, while we show that false positives are rare with our active approach.

***Index Terms*—** Data mining for forensic evidence

1. INTRODUCTION

While BitTorrent provides the ability to transfer files among many users quickly and efficiently, experience has shown that its decentralized architecture also makes it appealing for sharing copyright protected files illegally. With a peer-to-peer network like BitTorrent, content is distributed and replicated among a potentially large set of peers, making the process of finding and contacting each peer hosting the content in question a difficult task. Despite the challenge, entities acting on behalf of copyright holders have begun to monitor BitTorrent file transfers on a massive scale to identify and contact users who violate copyright laws.

In fact, a recent study [1] shows how the entities representing copyright holders use naïve techniques such as querying the BitTorrent tracker servers to identify individual users participating in an illegal file transfer. After being identified, these entities often distribute DMCA take-down notices or even pursue more formal legal sanctions against individuals who appear in the tracker's peer list. However, this simple approach is prone to a wide variety of errors. For instance, it is trivial to introduce erroneous information into the tracker lists by explicitly registering fake hosts to the tracker. The authors of the recent study demonstrate this type of false positive identification by registering networked devices such as printers and wireless access points to tracker lists and subsequently receiving DMCA take-down notices for their suspected participation in illegal file transfers.

This strategy of polluting tracker lists with fake peers could be used to frustrate anti-piracy investigations. The Pirate Bay, a popular tracker hosting site, has allegedly begun to inject arbitrary, but valid IP addresses into their tracker lists [2]. This counter-strategy may further increase the potential for false positive identification, which could have serious consequences as this evidence can be used to initiate legal action against suspected file sharers.

Given the inaccurate nature of the current techniques for monitoring BitTorrent file transfers and the clear need for effective anti-piracy tactics, we consider this question: Is it feasible to develop and deploy an efficient technique for identifying and monitoring peers engaged in file sharing that is more accurate than querying the trackers?

To answer this question, we propose a technique that is active, yet efficient. Starting with the tracker's peer lists, each peer listed by the tracker server is actively probed to confirm their participation in the file sharing and to collect concrete forensic evidence. Our tool, called BitStalker, issues a series of lightweight probes that provide increasingly conclusive evidence for the peers' active participation in the file sharing.

To evaluate the feasibility of this active approach in practice, we conduct a measurement study with real, large torrents. In particular, we quantify the number of peers that can be identified, the potential for falsely identifying peers, the potential for missing peers, and the cost associated with this technique in terms of bandwidth. Our results indicate that active probing can identify a sufficiently large portion of the active peers while requiring only 14.4–50.8 KB/s and about five minutes to monitor over 20,000 peers (using a commodity desktop machine). We also show that the active probing can be parallelized and scale to monitor millions of peers inexpensively using cloud computing resources such as Amazon's Elastic Compute Cloud (EC2) [3]. Using EC2, we estimate that our method can monitor the entire Pirate Bay (about 20 million peers) for only $12.40 (USD).

2. BACKGROUND

Before we describe our method for monitoring large BitTorrent swarms, we first provide a description of the BitTorrent protocol and an overview of the techniques currently being applied to identify peers who are sharing a file with BitTorrent.

2.1. The BitTorrent Protocol

To share a file, BitTorrent first breaks the file into several fixed size *pieces* and computes a SHA1 hash of each piece to verify integrity. Pieces are sub-divided into smaller data units called *blocks*, typically 16 KB in size. A metadata file containing the SHA1 hashes for each piece along with other information necessary to download the file including a URI to the *tracker server* is distributed to interested users via an out-of-band mechanism. Once a user has obtained the metadata for a file of interest, they proceed by contacting the tracker server to obtain a randomly chosen subset of peers who are sharing

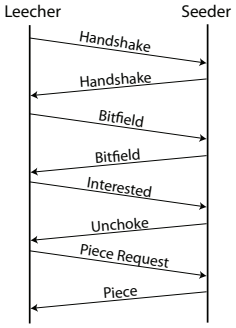

Fig. 1. BitTorrent message exchange to start a piece transfer

the file. This is called the *peer list*. By obtaining a peer list from the tracker (or another distributed hash table-based or gossip-based mechanism), the peer also registers itself with the tracker. The peer then begins requesting blocks of the file. Peers that are downloading pieces of the file are called "leechers," while peers that possess all pieces and participate as uploaders are referred to as "seeders."

The precise sequence of messages involved in the request of pieces is shown in Figure 1. A leecher establishes communication with another peer by exchanging `handshake` messages. The handshake consists of a plain text protocol identifier string, a SHA1 hash that identifies the file(s) being shared, and a peer identification field. After the handshake exchange, the leecher transmits a `bitfield` message. This contains a bit-string data structure that compactly describes the pieces that the peer has already obtained. After exchanging bitfields, the leecher knows which pieces the other peer can offer, and proceeds to request specific blocks of the file. The leecher sends an `interested` message to notify the other peer that it would like to download pieces. The other peer responds with an `unchoke` message only if it is willing to share pieces with the leecher. Upon receiving an unchoke message, the leecher asks for specific blocks of the file.

2.2. BitTorrent Monitoring Practices

While BitTorrent provides an efficient way to distribute data to a large group of users, it is also an appealing technique to distribute copyright protected files illegally. Copyright enforcement is particularly challenging within the context of BitTorrent, since the file(s) in question are distributed among a set of arbitrarily many peers. The copyright holders must first *identify* every user who appears to be sharing the file and ask them to stop sharing.

Despite the significant amount of work required to monitor BitTorrent networks, a recent study has gathered evidence showing that investigative entities acting on behalf of various copyright holders are monitoring and tracking BitTorrent users who are suspected of sharing copyright protected files [1]. These investigators — including BayTSP [4], Media Defender [5], and Safenet [6] who are hired by organizations such as the Motion Picture Association of America (MPAA) and the Recording Industry Association of America (RIAA) — are using *passive* techniques, such as querying the trackers for the peer lists to identify users who are engaged in illegal file sharing. Once a list of peers has been obtained, an ICMP echo (ping) message is sent to each IP address to ensure that it is alive.

However, as the aforementioned study notes, these methods for monitoring large BitTorrent networks can be wildly inaccurate. For instance, it is possible to implicate arbitrary networked devices by simply registering their IP addresses with the tracker server. In addition, *false positive* identification is also possible as a result of naturally occurring (*i.e.*, non-intentional) activity. For instance, the tracker may provide stale peer information, which may result in a user who recently obtained a DHCP lease on an IP address being implicated in the file sharing. The very real potential for false positives could have serious implications, since the investigators who conduct this monitoring often issue DMCA take-down notices or even initiate legal actions against the suspected file sharers.

3. ACCURATE AND EFFICIENT MONITORING

In order to study the feasibility of collecting forensic evidence to concretely prove a peer's participation in file sharing, we present *BitStalker*. BitStalker is active, yet efficient, since it consists of small probe messages intended to identify whether a peer is actively engaged in a file transfer. First, to obtain the list of peers who are potentially sharing the file, the tracker is queried. For each IP address and port number returned, we conduct a series of light-weight probes to determine more conclusively whether the peer really exists and is participating in the file transfer.

TCP connection. The first probe consists of an attempt to open a TCP connection to the IP address on the port number advertised by the tracker. A successful TCP connection indicates that the suspected peer is listening for connections on the correct port.

Handshake. If a TCP connection is established, a valid BitTorrent handshake message is sent. If the handshake succeeds, then the investigator has obtained evidence that the suspected peer is responding to the BitTorrent protocol, and may even provide information about the BitTorrent client software being used.

Bitfield. If the handshake probe succeeds, then a BitTorrent bitfield message is sent. This message contains a concise representation of all pieces that have been downloaded by the peer. A random bitfield is generated so that the probe looks like a valid bitfield message. If a peer responds with a valid bitfield message, then the investigator has obtained evidence that the peer has downloaded the part of the file that is described by their bitfield. This also indicates whether the peer is a seeder or a leecher. This provides the strongest form of forensic evidence that the peer is actively sharing the file without exchanging file data.

Block request. If the bitfield probe succeeds, we finally attempt to request a 16 KB block of the file from the peer. First, the peer's bitfield is examined to find a piece of the file that the peer has obtained. Next, this probe sends an interested message to indicate that we want to exchange pieces with this peer. The peer responds with an unchoke message, which implies that we are allowed to ask for pieces. We finally request a 16 KB block. If the peer responds with the block requested, then this probe succeeds. A single block is the smallest amount of data necessary to confirm that another peer is sharing the file. If the investigator has the remaining blocks of that piece, then they can verify the hash to ensure that the block is valid.

We argue that each probe type provides increasingly conclusive evidence of a peer's active involvement in file sharing. A successful TCP probe indicates that the peer is listening on the correct port. However, an effective counter-strategy could be to register arbitrary IP addresses with ports that are opened (such as web servers). The subsequent handshake probe is more conclusive, as it indicates that the BitTorrent protocol

Table 1. Summary of data sources

Torrent ID	Total Peers	Media Type
1	20,354	TV Series
2	16,979	TV Series
3	11,346	TV Series
4	14,691	TV Series
5	23,346	Movie
6	20,777	TV Series
7	24,745	TV Series
8	13,560	TV Series
9	19,694	TV Series
10	20,611	Movie
Total:	186,103	

is running on the correct port and also identifies the content being shared by a SHA1 hash. The bitfield probe provides stronger evidence still, since it describes all pieces that the peer has downloaded, which implies active sharing. Finally, requesting and subsequently receiving a block of the file provides the strongest form of concrete evidence for file sharing.

Practical considerations. The active probing framework can monitor peers who are actively participating in the file sharing. However, if a peer has just joined the torrent when they are probed, then they may not have any pieces of the file yet. Consequently, according to the BitTorrent protocol, if a peer has no pieces, then the bitfield probe is optional. Since the peer has not yet obtained any pieces of the file, the probing does not collect any evidence from this peer. If peers are probed repeatedly over time, then the likelihood of this case becomes negligible.

Additionally, "super-seeding" mode is enabled when a torrent is first established and there are few seeders. Super-seeding mode ensures that the original seeder is not overwhelmed by piece requests from other peers before it transfers data to another peer. When super-seeding is activated, the seeder may advertise an empty or modified bitfield, even though they possess every piece. Since we are interested in monitoring mature torrents consisting of at least tens of thousands of peers, we disregard new torrents in super-seeder mode.

Lastly, it is possible that peers may be able to detect the monitors and blacklist them. Siganos *et al.* show that the current passive BitTorrent monitors can be detected by observing that the frequency with which the monitor's IP addresses occur across a large number of tracker lists is statistically higher than that of normal peers [7]. Our active monitoring may also be identifiable in the same manner. To address this, we recommend that the monitoring be distributed across a large number or dynamic set of IP addresses.

4. EXPERIMENTAL EVALUATION

In this section, we present experiments to quantify both the effectiveness and the cost of monitoring large BitTorrent swarms using the active probing technique. In addition, we compare the accuracy, potential for false positives and false negatives, and the cost with the current strategy employed widely by anti-piracy investigators.

4.1. Data Sources and Methodology

To evaluate our light-weight probing technique, we selected ten large torrents each containing between 11,346 and 24,745 unique peers. In total, our experimental evaluation consists of over 186,000 peers. Peers participating in these torrents were sharing new theatrical releases and episodes of popular television shows (summarized in Table 1). These swarms represent the type of file sharing that may be monitored by copyright enforcement agencies.

To conduct the active probing, we wrote a tool called BitStalker that can perform the following tasks:

- Establish a TCP connection with another peer
- Exchange handshake messages with the correct SHA1 content hash and receive handshake responses
- Exchange bitfield messages and receive bitfield responses
- Request and receive a 16 KB block of file data

In short, BitStalker efficiently probes for participation in the BitTorrent protocol by sending and receiving a minimal number of small control messages rather than downloading the entire file from other peers.

The experiments were conducted as follows: The tracker server is contacted to obtain a subset of the peers who are currently believed to be sharing the file. Since the trackers only return a randomly selected set of 100 peers, it is necessary to query the tracker several times to obtain a large portion of the hosts registered with the tracker. Once peers are obtained from the tracker, BitStalker attempts to establish a TCP connection with each peer on its advertised TCP port. If a connection is established, a handshake message exchange is attempted. If handshake messages are exchanged, BitStalker attempts to exchange bitfield messages. Finally, if bitfields are exchanged, the tool attempts to retrieve a single block of the file. This procedure is repeated for each torrent to be monitored.

We compare our active probing method with the current approach to peer identification described in Section 2.2. After obtaining the list of suspected peers from the tracker, our tool sends precisely five ICMP echo (ping) messages to each IP address in the peer list. If a host responds to at least one ping, then it is assumed (perhaps erroneously) to be alive and sharing the file.

4.2. Experimental Results

We evaluate the proposed peer probing technique with regard to the number of peers that can be identified, an estimate of the number of peers that are falsely identified as being a file sharer (false positives), an estimate of the number of peers that this technique fails to identify (false negatives), and the measured cost of performing this active probing. The probing mechanism is compared along each of these metrics to the passive identification process using ping messages to verify the tracker's peer list.

Fraction of peers that respond. We first consider how many peers can be identified by active probing. As shown in Table 2, the fraction of peers that can be positively identified by each probe type increases with additional repetitions. To determine if additional peers can be identified through multiple probing attempts, the experiments are repeated ten times. Even though the number of peers probed remains constant for each repetition, we find that the fraction of peers that respond to probes increases, since some peers may be busy interacting with other peers when we probe.

The complete results for each torrent are given in Figure 2. Across the ten torrents, we could establish a TCP connection with between 26.7–44.6% of the peers listed by the tracker. While this percentage seems low, it is reasonable since many BitTorrent clients impose artificial limits on the number of open connections allowed, in order to reduce the amount of bandwidth consumed. A similar fraction of peers that establish connections is reported by Dhungel *et al.* [8].

The naïve ping method returns roughly the same fraction of peers as the active TCP connection probe. However, as we

Table 2. The average fraction of peers identified in one, five, and ten iterations of the monitoring across all ten torrents

Repetitions	Connection	Handshake	Bitfield	Block Request
1	30.8%	18.9%	17.7%	0.29%
5	35.9%	26.3%	25.3%	0.80%
10	36.9%	28.4%	27.6%	1.13%

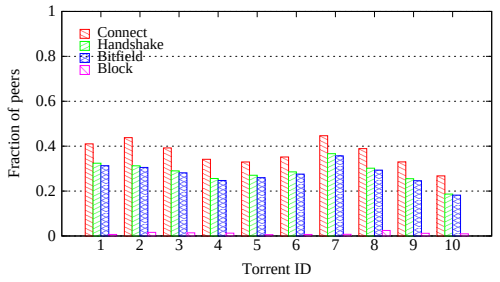

Fig. 2. Over ten runs, the cumulative fraction of peers identified with connections, handshakes, bitfields, and block requests across all ten torrents

will show, the ping probes are susceptible to an intolerably high number of false positives, while active probing significantly reduces the potential for false positives.

Both the handshake and bitfield probes succeed for between 18.6–36.6% of the peers. While this is lower than the TCP connection probe, it provides significantly stronger evidence for file sharing. For this fraction of the peers, an investigator can tell that the peer is obeying the BitTorrent protocol, sharing the correct file identified in the handshake probe by a SHA1 hash, and advertising the pieces of the file that the peer already possesses as identified in the bitfield probe. We argue that this small reduction in the fraction of peers that respond to bitfield probes is a small price for greater confidence in the identification results.

Finally, we observe that block request probes succeed for a very small faction of the peers, only 0.6–2.4%. This may be partly a result of BitTorrent's tit-for-tat incentive mechanism [9], which attempts to mitigate selfish leechers by enforcing reciprocity in the piece request process. This is implemented by uploading to other leechers from whom you download. The leecher with the highest upload rate receives download priority. Since BitStalker has a zero upload rate, it does not receive priority for piece requests. However, BitTorrent does offer optimistic unchoking, which enables a leecher to download regardless of their upload rate. BitStalker only receives pieces from other peers who have chosen to optimistically unchoke.[1] Since only about 1% of the peers respond to our block requests on average, we argue that the minimal additional evidence obtained through this probe is not worth the extra time and bandwidth required to collect this evidence.

False positives. The most serious flaw with the past and present investigative tactics based on tracker list queries and ping probes is the real potential for a high number of false positives. Furthermore, active peer list pollution further increases the potential for false positives.

To establish a lower bound on false positives obtained by the naïve investigative strategy, we count the number of peers that respond to pings yet show no indication of running any network service on their advertised port. More technically, if a peer responds to a TCP SYN request with a TCP RST (reset) packet, this indicates that the remote machine exists, but it is not running any service on the advertised TCP port. From our experiments, we observe that 11% of peers exhibit this behavior on average and are, therefore, definite false positives using this naïve investigative strategy.

In addition, we count the number of peers that *could* be false positives with the ping method. These are the peers that respond to ping probes, but ignore the TCP probe (*i.e.,* no connection or reset packet). From our experiments, we find that on average an additional 25.7% of the peers could potentially be false positives, but we cannot say this conclusively. It's possible that some of these peers could have reached a connection limit in their BitTorrent client or could be filtering incoming traffic.

In contrast to the naïve ping method, the active probing strategy offers more reliable peer identification with few avenues for false positives. For instance, a successful TCP probe indicates that the peer is listening for connections on its advertised port. However, one could envision a more intelligent pollution strategy where arbitrary IP addresses with open ports are inserted into trackers (*i.e.,* real HTTP or FTP servers). The subsequent handshake and bitfield probes would then eliminate this form of pollution by checking that the host is running the BitTorrent protocol.

However, the active probing approach is not entirely immune from the possibility of false positive identification. For example, peers using an anonymizing network such as Tor [10] may produce false positives, since the last Tor router on the client's path of Tor routers (called a Tor exit router) would be implicated in the file sharing. In fact, a recent study has found that BitTorrent is among the most common applications used with Tor [11].

To determine how common this type of false positive is in practice, we compare the list of potential BitTorrent peers obtained through our experiments to the list of all known Tor exit routers provided by Tor's public directory servers. On average, we find that only approximately 1.8% of the peers are using Tor to hide their identities.[2] However, these are not false positives using active probing, since a peer using Tor (or another anonymizing network or proxy service) cannot bind to the advertised port on the exit host to accept incoming connections. Consequently, active probing does not provide any evidence for these peers. Furthermore, peers using Tor are easily identifiable and can be filtered out of the results.

In addition to general-purpose anonymizing networks, solutions have been proposed specifically for anonymizing BitTorrent. For instance, SwarmScreen's goal is to obscure a peer's file sharing habits by participating in a set of random file sharing swarms [12]. Also, BitBlender attempts to provide plausible deniability for peers listed by the trackers by introducing relay peers that do not actively share files, but rather act as proxies for other peers actively sharing the file [13]. The active methods we propose would identify peers utilizing SwarmScreen and BitBlender as file sharers. While these peers are not intently sharing content, an investigator may still be interested in pursuing these peers since they contribute pieces of the file to other peers who are actively sharing.

False negatives. False negative identification occurs when a peer who is actively sharing a file cannot be identified as a file sharer. Both the active probing technique and the naïve ping method suffer from the potential for false negatives. The ping method may miss peers who are behind a firewall that blocks incoming ICMP traffic. For example, this is the default configuration for Windows Vista's firewall settings. The active probing method may also suffer from false negatives when a

[1] Additional blocks may be received if BitStalker offered blocks before asking for blocks.

[2] However, several peers could be using each of these Tor exit nodes.

Table 3. Size of each probe type (assuming no TCP options)

Probe Type	Description	Size
TCP connection	Three-way handshake	162 Bytes
Handshake	Handshake request/reply	244 Bytes
Bitfield	Bitfield request/reply	Variable
Block Request	Block request/reply	16.688 KBytes
ICMP Ping	Ping request/reply	86 Bytes

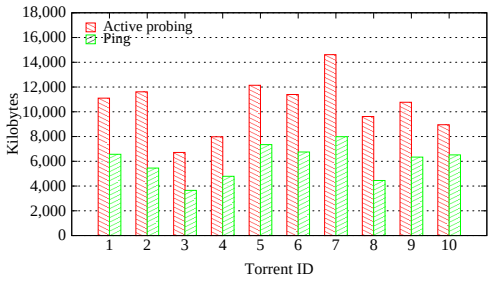

Fig. 3. Total amount of traffic necessary to monitor each torrent using active probing and pings

peer's number of allowed connections is at the maximum. In this case, the initial TCP connection probe will fail to identify that the peer is listening on its advertised port. In general, we found that repeating the monitoring procedure decreases false negatives. Table 2 shows that the number of false negatives decreases as the experiment is repeated. Although there are diminishing returns, as the false negatives do not decrease significantly between 5 and 10 iterations of the monitoring.

We can, however, provide a lower bound on false negatives obtained with the naïve ping method. This is achieved by counting the number of peers that do not respond to pings, but do respond to the TCP connection probe. Our experiments show that the naïve ping method would fail to identify at least 22.3% of the peers on average.

Cost. In order for an active probing strategy to be a feasible technique to monitor large BitTorrent swarms in practice, it is necessary for the probing to be as efficient as possible. Table 3 shows that the size of each probe is small and Figure 3 shows the amount of traffic that was required to monitor each torrent using the active probing technique. For comparison, the cost for the ping method is also plotted. While the ping approach requires less bandwidth, we have shown that it is not sufficiently accurate in identifying active file sharers. Using a modest Linux desktop machine, it took 304.5 seconds on average to monitor an entire torrent, which required only 14.4–50.8 KB/s of bandwidth. The active probing overhead is dependent on the fraction of peers that respond to active probes. This is an intuitive result, implying a direct relationship between the number of peers identified and the amount of bandwidth required by the probing.

The active probing method is also highly scalable, particularly when inexpensive cloud computing resources such as Amazon's Elastic Compute Cloud (EC2) [3] are utilized. Machines from EC2 are available at a small cost dependent on the execution time and bandwidth usage of the jobs. From our experiments, on average we probed approximately 61 peers/second, uploaded 288.2 bytes/peer and downloaded 296.6 bytes/peer. Using EC2's pricing model, we estimate that it is possible to monitor peers at an expected cost of roughly 13.6 cents/hour (USD). In fact, it's possible to scale the active probing to monitor the entire Pirate Bay, which claims to track over 20 million peers [14]. We estimate that this method can monitor the Pirate Bay for $12.40 (USD).

5. CONCLUSION

This paper presents *BitStalker*, a low-cost approach to monitoring large BitTorrent file sharing swarms. BitStalker collects concrete evidence of peers' participation in file sharing in a way that is robust to tracker pollution, highly accurate, and efficient. In contrast, the past and present investigative monitoring strategy consists of tracker server queries and ICMP ping probes. While this method is simple, it is also prone to a variety of significant errors, especially false positive identification, since this monitoring technique does not verify participation in the file sharing. We present an alternative monitoring strategy based on actively probing the list of suspected peers to obtain *more conclusive* evidence of participation in the file sharing.

There are several aspects of our approach that warrant additional attention. In particular, a specific definition of what constitutes "evidence" in the context of file sharing across various legal systems should be explored. Also, the general legal issues that this type of monitoring exposes should also be investigated further.

Acknowledgments. The authors thank the anonymous reviewers for their valuable comments and suggestions. We also thank Claire Dunne and the University of Colorado's institutional review board for ensuring that this research was conducted with the highest of ethical standards. This research was funded in part through gifts from PolyCipher.

6. REFERENCES

[1] Michael Piatek, Tadayoshi Kohno, and Arvind Krishnamurthy, "Challenges and directions for monitoring P2P file sharing networks – or – Why my printer received a DMCA takedown notice," in *3rd USENIX Workshop on Hot Topics in Security (HotSec)*, July 2008.

[2] "Pirate bay tricks anti-pirates with fake peers," `http://torrentfreak.com/the-pirate-bay-tricks-anti-pirates-with-fake-peers-081020`.

[3] "Amazon elastic compute cloud (amazon ec2)," `http://aws.amazon.com/ec2`.

[4] "BayTSP," `http://www.baytsp.com`.

[5] "Media defender – P2P anti-piracy and P2P marketing solutions," `http://www.mediadefender.com`.

[6] "Safenet Inc: The foundation for information security," `http://www.safenet-inc.com`.

[7] Georgos Siganos, Josep M. Pujol, and Pablo Rodriguez, "Monitoring the Bittorrent monitors: A bird's eye view," in *PAM*, 2009, pp. 175–184.

[8] Prithula Dhungel, Di Wu, Brad Schonhorst, and Keith W. Ross, "A measurement study of attacks on bittorrent leechers," in *International Workshop on Peer-to-Peer Systems (IPTPS)*, February 2008.

[9] "BitTorrent protocol specification," `http://wiki.theory.org/BitTorrentSpecification`.

[10] Roger Dingledine, Nick Mathewson, and Paul Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[11] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker, "Shining light in dark places: Understanding the Tor network," in *Proceedings of the 8th Privacy Enhancing Technologies Symposium*, July 2008.

[12] David R. Choffnes, Jordi Duch, Dean Malmgren, Roger Guierma, Fabian E. Bustamante, and Luis Amaral, "SwarmScreen: Privacy through plausible deniability for P2P systems," Northwestern EECS Technical Report, March 2009.

[13] Kevin Bauer, Damon McCoy, Dirk Grunwald, and Douglas Sicker, "BitBlender: Light-weight anonymity for BitTorrent," in *Proceedings of the Workshop on Applications of Private and Anonymous Communications (AlPACa 2008)*, Istanbul, Turkey, September 2008, ACM.

[14] "The pirate bay," `http://thepiratebay.org`.