BitBlender: Light-Weight Anonymity for BitTorrent

Kevin Bauer, Damon McCoy, Dirk Grunwald, and Douglas Sicker
Department of Computer Science
University of Colorado
Boulder, CO 80309-0430 USA
{bauerk, mccoyd, grunwald, sicker}@colorado.edu

ABSTRACT

We present *BitBlender*, an efficient protocol that provides an anonymity layer for BitTorrent traffic. BitBlender works by creating an ad-hoc multi-hop network consisting of special peers called “relay peers” that proxy requests and replies on behalf of other peers. To understand the effect of introducing relay peers into the BitTorrent system architecture, we provide an analysis of the expected path lengths as the ratio of relay peers to normal peers varies. A prototype is implemented and experiments are conducted on Planetlab to quantify the performance overhead associated with the protocol. We also propose protocol extensions to add confidentiality and access control mechanisms, countermeasures against traffic analysis attacks, and selective caching policies that simultaneously increase both anonymity and performance. We finally discuss the potential legal obstacles to deploying an anonymous file sharing protocol. This work is among the first to propose a privacy enhancing system that is designed specifically for a particular class of peer-to-peer traffic.

Categories and Subject Descriptors

C.2.0 [**Computer Systems Organization**]: Computer-Communication Networks—*Security and protection*;
C.2.2 [**Computer Systems Organization**]: Computer-Communication Networks—*Applications*

General Terms

Design, Legal Aspects, Performance, Security

Keywords

Anonymity, BitTorrent, Peer-to-peer networks

1. INTRODUCTION

General purpose, low latency anonymous networks are currently being used to provide private and anonymous communication services for a variety of applications. For instance, Tor [11] has become the standard tool for anonymizing TCP traffic. This is largely because of its ability to provide low latency anonymous transport to facilitate interactive applications such as web browsing and instant messaging. However, it has been noted that Tor’s ability to provide a low latency anonymous transport service is being potentially threatened by the excessive amount of peer-to-peer (P2P) file sharing traffic that it transports [15]. Since there is a clear demand for anonymous file sharing and to alleviate the strain placed upon the Tor network by this P2P traffic, we present the design and implementation of an anonymous network protocol specifically tailored to P2P file sharing, and in particular, the BitTorrent protocol.

BitTorrent ostensibly provides no built-in support for anonymous file sharing. In fact, as part of the default peer discovery method, the protocol requires that the IP addresses of all peers sharing the file be published by a well-known and publicly accessible server called a tracker. By querying the tracker, it is currently trivial to determine who is actively sharing a particular file. In fact, a study found that information from these trackers is currently being used to identify file sharers, often with poor accuracy [18]. Hence, it is not surprising that many BitTorrent users resort to using Tor to remain anonymous.

Traditionally, *mix networks* have been the fundamental building block for many privacy enhancing systems. Mix networks construct a chain of intermediate hops between the source and destination of a message to conceal the message’s true sender and receiver. Batching of messages and cover traffic are common techniques to further frustrate traffic analysis attempts. Most mix networks attempt to provide a high degree of anonymity, suitable for protecting cyber-dissidents in countries where Internet freedoms are not protected. In this case, the strongest practical anonymity available is required.

In contrast, it is often acceptable to provide a lower degree of anonymity. Reiter and Rubin [19] describe *degrees of anonymity* as a spectrum that expresses the confidence that an adversary has regarding the identity of the real initiator of a message. Their system, Crowds, achieves varying degrees of anonymity for web transactions by routing each message through a set of intermediate hops in a probabilistic fashion. When a message is received by an intermediate node, it is either forwarded to another intermediate hop or delivered to its final destination with a certain pre-defined probability. From the perspective of the destination server, it is unclear whether the node from which it received the

SecureComm’08, AIPACa Workshop, September 22, 2008, Istanbul, Turkey
Copyright 2008 ACM 978-1-60558-241-2 ...$5.00.

message is the initiator or a proxy for another node. Thus, all nodes that participate in such a network enjoy a certain degree of plausible deniability with regard to requesting a file.

Inspired by Crowds, we propose a similar low-overhead anonymity layer for BitTorrent that offers sufficient anonymity properties to achieve the condition of plausible deniability. To this end, we present BitBlender, an anonymity layer for the BitTorrent protocol that has low overhead and provides varying degrees of anonymity. BitBlender achieves plausible deniability by introducing special "relay peers" that forward data on behalf of the peers that are actively sharing a file. When peers request pieces of a file, it is difficult to determine whether a piece is delivered by another peer engaged in the file transfer, or if it is delivered through one or more relay peers. Thus, the degree of anonymity provided is dependent upon the number of relay peers relative to the number of normal peers participating in a file transfer; however, the expected performance overhead is also dependent upon the number of relay peers participating, as the path length between the initiator and the responder is higher on average as more relay peers participate.

BitBlender is among the first to explore light-weight privacy enhancing system designs without the use of cryptography. Strict data confidentiality within BitTorrent is unnecessary, since files are typically shared publicly and anyone can participate without requiring any special access or authorization. In addition, this protocol has the ability to provide a level of anonymity that is tunable, so it can be adjusted for the sensitivity of the data transferred. Finally, since BitBlender requires no modifications to the existing BitTorrent protocol, it is easy to deploy within BitTorrent's current system architecture.

We provide an analysis of the protocol in terms of its expected path length as the ratio of normal peers to relay peers varies. In addition, we show that BitBlender has a lower hop count on average than Tor, even when the ratio of relay peers to normal peers is greater than 1/2. To analyze the anticipated performance overhead, we implement a prototype and perform experiments to quantify the expected additional download time that will be experienced by the end users as the number of participating relay peers varies. We also compare BitBlender's expected performance to that of BitTorrent tunneled over Tor.

Having presented the basic protocol, we present extensions aimed at strengthening the anonymity and increasing the performance. A confidentiality and access control mechanism is detailed that would provide confidential and authenticated file transfers. Also, we address traffic analysis attacks and present simple countermeasures. We lastly explore selective caching as a mechanism to simultaneously impede certain traffic analysis tactics and decrease the expected path length.

Finally, we discuss some of the legal questions that BitBlender and other general purpose anonymous networks present. In particular, the legality of operating an open relay is unclear, and arguably the continued success of anonymous communication systems relies on policy makers from around the world providing some form of legal protection for the operators of anonymous networks.

The remainder of this paper is organized as follows: In Section 2 we provide a high level overview of the traditional BitTorrent protocol and a discussion of how varying degrees of anonymity are defined. Section 3 presents the primary design principles behind BitBlender. In Section 4, we define the BitBlender protocol and provide an analysis in Section 5. In Section 6, the performance of a real prototype implementation is analyzed. We present extensions to the basic protocol in Section 7 and discuss the potential legal implications in Section 8. A survey of related work is given in Section 9 and concluding remarks are provided in Section 10.

2. BACKGROUND

In this section, we describe the traditional BitTorrent protocol in sufficient detail to understand how our proposed anonymity layer fits into the standard protocol. In addition, we define the condition of plausible deniability, and how that fits within the anonymity spectrum as defined in Crowds.

2.1 The BitTorrent Protocol

BitTorrent was initially designed with the goal of providing a decentralized content distribution network based upon a swarming peer-to-peer model. Files are distributed by breaking the original file into several fixed-size *pieces*. Peers may host arbitrary content to share with other peers in the network by creating a metadata file called a *torrent*, which contains the information necessary to download a file. This includes the number of pieces, the hashes of each piece in a hosted file, and a pointer to a server called a *tracker*. The tracker maintains a list of the peers associated with the given torrent. Peers that have a complete copy of the file to share are called *seeders*, and those peers that download are *leechers*.

To download a file using BitTorrent, using the torrent file associated with the file, the peer contacts the tracker and retrieves a list of other peers that are currently uploading or downloading the file. The peer then issues requests for pieces from other peers. When a complete piece is received, the peer verifies the integrity using the hash published in the torrent file. Once all of the pieces of a file are downloaded, the peer may remain and provide pieces to other peers.

In addition to the centralized torrent model described above, BitTorrent has added a distributed tracker capability based upon a distributed hash table (DHT) that can be used en lieu of the tracker server.

BitTorrent based upon the traditional centralized tracker lacks the ability to provide anonymous file transfers since every peer associated with a particular torrent is either actively downloading or uploading the file. Some have argued that the DHT tracker offers a certain degree of anonymity for peers; however, despite the fact that not every peer is listed in a particular DHT query response, every peer that is listed is sharing the file. We propose the addition of an anonymity layer for the centralized tracker protocol that provides a degree of plausible deniability for peers; however, this anonymity layer may be extended to the DHT tracker model.

2.2 Degrees of Anonymity

In order to describe our anonymity layer for BitTorrent, it is necessary to define the notion of anonymity that the protocol provides. Reiter and Rubin describe anonymity as a spectrum, with degrees ranging from "absolute privacy" to "provably exposed" [19]. Between these extremes, the level of anonymity varies between states of "probable" and "possible" deniability. Probable deniability exists when an adversary can determine with a probability $0.5 \leq p < 1$ that a message in the system originated at a particular user. Possible deniability is the state at which there is a probability $0.5 > p > 0$ that a message originated at a specific user. We define *plausible deniability* as the state that encompasses both probable and possible deniability ($1 > p > 0$). The specific probability is precisely the ratio of relay peers to total peers (both relay and normal peers). With no additional information, an adversary has a probability p of correctly guessing whether an individual peer is a relay or a normal peer.

In addition to these degrees of anonymity, other anonymity metrics have been proposed. Pfitzmann and Waidner developed a concept of anonymity as being indistinguishable from within a set of possible identities [17, 24]. This is commonly referred to as k-anonymity, where k refers to the number of identities in the set. In this model, an adversary should ideally have a probability of no greater than $1/k$ of determining the true identity of an entity. In addition, Diaz *et al.* [10] and Serjantov and Danezis [21] proposed the use of entropy to measure the amount of information that can be ascertained through traffic analysis. However, we do not apply k-anonymity or information theoretic metrics in the subsequent analysis of BitBlender.

3. DESIGN PRINCIPLES

In order to describe the BitBlender protocol, it is necessary to first explain the design goals and the envisioned threat model.

3.1 Design Goals

BitBlender's design achieves the following:

- **Low overhead:** The protocol should be more lightweight in terms of computational resources, and should provide better throughput and lower latency in comparison to general-purpose anonymity networks. There is no overhead for cryptographic operations and protocol overhead associated with potentially routing messages through multiple relay peers is minimal.

- **Usability:** An important goal is usability, which means that the protocol should be easy to use, work seamlessly with the existing BitTorrent architecture, and offer performance that is comparable to – or better than – general-purpose anonymous networking protocols such as Tor. Performance is considered to be fundamental to the system's adoption and usability, since end users will be unlikely to use BitBlender if other systems such as Tor provide better performance.

- **Plausible deniability:** The protocol provides plausible deniability for peers that are listed by the tracker for a particular torrent. This is achieved by introducing relay peers that do not initiate file downloads or uploads, but simply proxy requests on behalf of other peers. By introducing relay peers, it is no longer the case that every peer in the tracker list is actively initiating uploads or downloads. An adversary must now engage in more sophisticated and potentially error-prone traffic analysis techniques to determine the true initiators. A detailed discussion of such traffic analysis attacks is provided in Section 5.3.

- **Tunable anonymity:** It is well-known that there is always a trade-off between the anonymity that a system can provide and its performance. BitBlender allows the trade-off between performance and anonymity to be made by tuning a system parameter, specifically the number of relay peers participating relative to the number of normal peers. This is an important feature, since some torrents may be more sensitive than others.

3.2 Threat Model

We assume a non-global adversary that can participate in the BitBlender protocol as a colluding fraction of the total peers (either relay or normal). This implies that the adversary can see the traffic flowing through the subset of the peers that it controls. In addition, the adversary can monitor the tracker list to see which other peers are participating in the torrent. This threat model is the same as that which is assumed in other low-latency anonymous networks [8, 11, 12, 19, 20]. We further assume that the adversary cannot passively monitor arbitrary links between peers in the network.

4. THE BITBLENDER PROTOCOL

Building upon the notion of anonymity provided by Crowds, we present BitBlender, an anonymity layer for BitTorrent. Before giving a high-level overview of the protocol, it is necessary to define each component. As in traditional BitTorrent, there are *peer* nodes that wish to share content. We introduce *relay peers* as peers that do not initiate downloads or uploads, but simply proxy traffic on behalf of normal BitTorrent peers. Relay peers and anonymous torrents are organized by an entity called a *blender*, which could be a single directory server, a set of directory server replicas, or a DHT.

The protocol proceeds as follows: In order to attract relay peers, the tracker for an anonymous torrent contacts the blender and requests that relay peers join the torrent with a certain probability. Given the degree of anonymity desired, the tracker asks each relay peer to probabilistically join its torrent.

Once the relay peers have joined the torrent, they proceed by transparently accepting piece requests and *forwarding* them to another member of the torrent. This peer may, in fact, be another relay peer, or it may be a real peer participating in the file transfer. Replies are also transparently forwarded in the same manner along the same relay path. Thus, an ad-hoc relay network is created, where the path lengths are somewhat non-deterministic. The relay peers could appear to be seeders, or they could advertise only a subset of the pieces for a particular file. The protocol's system architecture is described pictorially in Figure 1.

4.1 Relay Peer Joining

Let N be the set of peers participating in an anonymous torrent and M be the set of relay peers participating in the anonymous torrent such that $M \cap N = \emptyset$. The set of all relay peers listed by the blender is B, such that $M \subseteq B$. To establish an anonymous torrent, it is necessary that the tracker request a subset of the relay peers to join the anonymous torrent. The request sent by the tracker to the blender consists of the tuple (n, t), where n is the number of relay peers requested and t is a unique identifier for the tracker (such as a URI). Upon receipt of this message, the

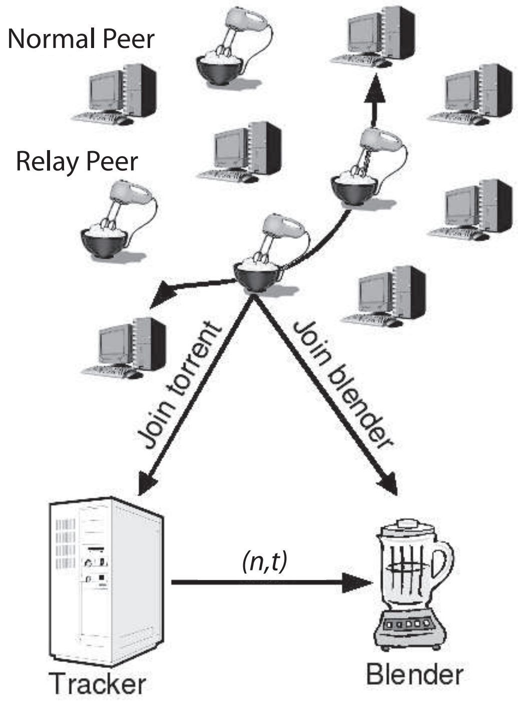

Figure 1: The BitBlender protocol system architecture. The protocol proceeds as follows: (1) A relay peer joins the blender; (2) The tracker requests relay peers; and (3) Relay peers probabilistically join the torrent. A piece request through two relay peers is shown (the path length is three hops).

blender must calculate a join probability p, based upon the number of nodes requested and the size of the relay peer set, where $p = \frac{n}{|B|}$. This enables the blender to remain agnostic about which relay peers join the torrent.

Each $b_i \in B$ chooses a pseudorandom number $r \in \mathbb{R}$ subject to $0 \leq r \leq 1$ and joins the torrent identified by t iff $r \leq p$. On average, the requested number of relay peers n will join the anonymous torrent.

4.2 Anonymity Layer

In order to provide an anonymity layer, the normal peers simply make requests as in the traditional BitTorrent protocol; however, if a relay peer is requested for a piece, the original request is forwarded to another peer, potentially another relay peer.

An ad-hoc relay network is constructed in this manner, where the path lengths are probabilistically influenced by the concentration of relay peers to real peers in the torrent. As the piece request reaches a real peer, it fulfills the request by sending the requested piece back through the chain of relay peers to the original requesting peer. A certain degree of anonymity is achieved since it is difficult to prove which peers in the torrent are relay peers and which are real peers.

4.3 Discussion

BitBlender is a low-overhead, usable and inter-operable anonymity layer for BitTorrent that provides a dynamically tunable level of plausible deniability. Plausible deniability is achieved by adding relay peers, since it is no longer trivial to infer the set of peers participating in a particular torrent simply by inspecting the peer list maintained by the tracker.

It is important that relay peers not only appear in the tracker list, but also forward requests and replies. If an adversary participates in the protocol, it would be relatively easy to determine which peers are actively participating in the torrent and which do not issue piece requests or replies. Thus, in order to provide a higher degree of anonymity, it is essential that the relay peers appear to be actively participating.

By allowing the tracker to explicitly specify the number of relay peers that should join, this allows individual torrents to have a tunable anonymity parameter. The more relay peers that join a torrent, the more difficult it would be for an adversary to determine the true set of peers participating in the transfer of a torrent.

BitBlender is fully inter-operable with the existing BitTorrent protocol. Peers that wish to obtain a degree of anonymity may participate in BitBlender; however, those peers that do not desire anonymity may still participate in the torrent. In this case, they would be easily identified as normal peers, since they do not appear in the blender. Since inter-operability is a design goal, we do not provide any confidentiality or access control mechanisms. Such a layer would disallow non-BitBlender peers from participating in the torrent. We do, however, explore data confidentiality and access control as an extension in Section 7.

BitBlender's ability to provide an anonymity layer without the use of expensive cryptographic operations is unique when compared to previous mix and onion routing-based anonymity systems. BitTorrent is a protocol whose content does not typically leak personal information like HTTP or instant messaging protocols [6, 9]. Thus, it is not essential to provide strong data confidentiality, since the contents of the torrent are easily accessible to anyone.

Finally, there are several considerations that must be weighed when designing the blender. If the blender is a single centralized directory server, it becomes a single point of failure in the system and is open to Denial of Service (DoS) attacks. One solution may be to simply replicate the blender's database throughout the network and employ a consensus technique to issue queries. This is more fault-tolerant, but is susceptible to Byzantine faults [14]. Finally, the blender may exist as a service accessible via a distributed hash table (DHT). In this case, the blender is fully distributed; however, simple DHT schemes and other gossip protocols can be targeted with Eclipse attacks [23]. Designing a distributed and secure directory service is a challenging problem. For the sake of simplicity, we assume a blender based upon a single centralized directory server.

5. PROTOCOL ANALYSIS

In this section, we analyze BitBlender in terms of expected path lengths, a comparison to Tor, and the potential for traffic analysis attacks.

5.1 Expected Path Length

Since the protocol forwards requests and replies in a probabilistic fashion dependent upon the number of relay peers participating, we present an analysis of the expected path length. For simplicity, we assume that peers are chosen for piece requests uniformly at random from the set of all participating peers. Formally, let N be the set of peers (both relay and normal) associated with an anonymous torrent; the probability of choosing an arbitrary peer $p_i \in N$ is $1/|N|$.

Let $M \subseteq N$ be the set of relay peers participating in the torrent and $P \subseteq N$ be the set of normal peers subject to $M \cap P = \emptyset$ and $|M| + |P| = |N|$. The path length l for a piece request from peer $p_i \in P$ through relay peers $M_l \subseteq M$ to $p_j \in P$, the peer satisfying the piece request, is dependent upon the ratio of relay peers to total peers in a torrent. This ratio is defined as $r = |M|/|N|$. Thus, the expected path length $E[l]$ is defined as an infinite geometric series:

$$E[l] = \sum_{i=0}^{\infty} r^i = 1 + r + r^2 + r^3 + \cdots + r^\infty = \frac{1}{1-r} \qquad (1)$$

subject to $0 \leq r < 1$.

When the ratio of relay peers to total peers is 0, the expected path length is 1.0. There is no relay overhead, since the peers are communicating directly (i.e., $M_l = \emptyset$). When the relay peers are 1/4 of the total peers, the expected path length is 1.33, as the relay peers rise to 1/2 of the torrent, the expected path length is 2.0, and as the relay peers out number total peers as 3/4 of the torrent, the expected path length is 4.0 hops.

5.2 Comparison to Tor

As stated in Section 4, BitBlender relies on the formation of ad-hoc paths to relay requests and replies. On the other hand, Tor establishes source-routed circuits by choosing a set of precisely three Tor routers (by default) and transporting TCP traffic through these routers using a layered encrypted scheme before the traffic reaches its final destination. By building source-routed circuits, the protocol ensures a path length of precisely four hops from the initiating client to the destination server. This provides relatively strong anonymity properties at the cost of lower throughput and higher latency on average. BitBlender offers a lower expected path length for anonymous torrents in which relay peers constitute less than 3/4 of the total peers participating in the file transfer.

In addition to the relatively high path length, Tor incurs additional protocol overhead to establish these circuits. This consists of layered encryption applied to the circuit-building messages and data packets in a fashion based on onion routing [13]. These circuit building messages must be sent whenever the client chooses to build a new circuit.

Finally, since all traffic is routed through potentially malicious Tor routers, strong confidentiality must be ensured to protect the traffic from local eavesdroppers. However, strictly speaking, the final Tor router that forwards the traffic to the destination server removes the final layer of encryption and can examine a user's payload.

BitBlender is unique in its ability to provide an anonymity service with minimal protocol overhead. Since content is publicly available and specific to each torrent, it is not a strict requirement that BitBlender provide confidentiality and access control (although we do provide these mechanisms as an extension to the protocol in Section 7.1).

5.3 Security Analysis

Recall that the primary threat model that this protocol should protect against is that of a non-global adversary that participates in the protocol but cannot monitor arbitrary links. Within this model, there exist attacks through which an adversary may gain information about users by recording traffic that it observes during its participation. A naïve attacker may attempt to request pieces through a peer to determine if they are, in fact, a normal peer. This simple attack would be unsuccessful, since relay peers and normal peers both appear to issue and fulfill piece requests.

More intelligent strategies could potentially gain information about the set of real peers. Over time, if an adversary observes that a peer makes a request for the same pieces multiple times, they may be identified as a potential relay [26]. To mitigate this type of attack, normal peers can issue the same piece requests multiple times in a nondeterministic fashion to appear indistinguishable from the relay peers. This technique can be regarded as a form of cover traffic, which is a well-studied traffic analysis mitigation strategy within the context of mix networks. Additionally, relay peers could cache previously requested pieces, and thereby exhibit more normal (and less distinguishable) behavior. Additional traffic analysis countermeasures are provided in Section 7.2.

In addition, Reiter and Rubin identify a set of timing attacks in which an intermediate node (i.e., a relay peer) can determine if the previous node on the path is the initiator of a request based upon an analysis of the time that elapses until the request is fulfilled [19]. If the time is sufficiently small, then the intermediate node can conclude with a certain level of confidence that the preceding node is the initiator. This is an instance of what Wright *et al.* call the predecessor attack [27]. BitBlender, like Crowds, is vulnerable to the predecessor attack. To address this threat, we propose that random delays and selective caching mechanisms be applied to perturb the timing of piece requests and responses (see Sections 7.2 and 7.3 for a discussion of these techniques). BitBlender's key accomplishment is that an adversary cannot determine which peers are sharing the file simply by examining the tracker; the adversary must now expend more resources and conduct traffic analysis.

6. PERFORMANCE ANALYSIS

In this section, we provide an analysis of the expected performance overhead for the BitBlender protocol in terms of download time as the number of relay peers varies. We also provide a performance comparison to BitTorrent tunneled over the Tor network.

6.1 Experimental Setup

In order to quantify the protocol's performance overhead, we implemented BitBlender's relay peers using the Enhanced CTorrent BitTorrent client [3]. To ensure that the performance evaluation is conducted in a realistic environment, we perform experiments using nodes from the PlanetLab testbed [16]. In these experiments, there are precisely three seeders, one centralized tracker hosting the torrent metafile for a 1 MB file, and 20 normal peers actively sharing the file. The file is distributed in 1 KB pieces. We emulate resource-constrained peers, such as those behind an asymmetric residential cable modem link. All peers are limited to 1 MB/s for downloads and 256 KB/s for uploads. To understand the effect of introducing relay peers into the network, we conduct experiments by adding 5, 10, 15, and 20 relay peers to the network. Each experiment is repeated three times to compute statistics.

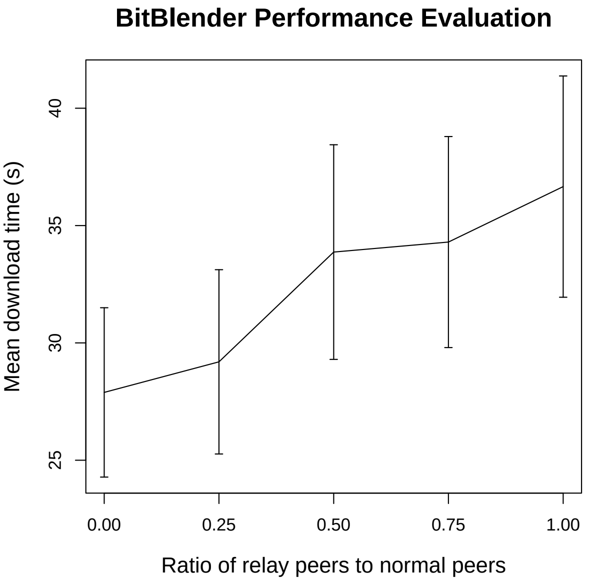

Figure 2: Mean download time with 95% confidence intervals as a function of the ratio of relay peers to normal peers.

In order to quantify the performance improvement that BitBlender offers, we provide a performance evaluation of a popular method for sharing files anonymously: BitTorrent run over the Tor network. In this experiment, there are 20 seeders a single peer using the Azureus BitTorrent client [2] tunneled over Tor version 0.2.0.30 (from August 2008). The peers share the same 1 MB file and are rate-limited as described above. This experiment is repeated ten times.

6.2 Experimental Results

We first analyze BitBlender's performance in terms of the expected download time as the ratio of relay peers to normal peers varies. We next compare BitBlender's expected download times to that of BitTorrent over the Tor network.

6.2.1 Adding Relay Peers

As shown in Figure 2, the mean download time across all 20 peers steadily increases with the number of relay peers participating in the anonymous torrent. As a baseline, when no relay peers participate, peers download the file in approximately 27.9 seconds on average. In the worst case when the ratio of relay peers to normal peers is 1.0, the mean download time is about 36.7 seconds. Note that the download time increases only minimally as more relay peers are added. Since BitTorrent is by its nature a swarming protocol, the performance degradation introduced by having more relay peers participate is partially masked by BitTorrent's tendency to download and upload pieces from multiple peers simultaneously. Thus, the protocol offers reasonably high performance even as the ratio of relay peers to normal peer is relatively high.

6.2.2 Comparison to Tor

Over the course of the ten experiments, the mean download time is 215.1 seconds with a 95% confidence interval of 199.8–230.4 seconds. Not surprisingly, the expected download time using Tor for anonymity is significantly higher than BitBlender, even when the ratio of relay peers to normal peers is 1.0.

7. PROTOCOL EXTENSIONS

Having presented the basic BitBlender protocol and analysis, we now focus on optional extensions to strengthen the anonymity and increase the performance.

7.1 Confidentiality and Access Control

While the attack model assumes that an adversary cannot monitor arbitrary links, it might be the case that an ISP or set of ISPs collude with the adversary. In this case, a confidentiality and access control mechanism would offer an increased level of privacy and anonymity. To this end, we present an additional confidentiality and access control layer. In addition to its role as a directory for relay peers, the blender should also sign public keys for both relay and normal peers as a trusted authority. This public key infrastructure (PKI) can be used by peers to authenticate each other and to restrict access to the content in the torrent. Once authenticated, peers can establish encrypted tunnels using a protocol such as Transport Layer Security (TLS) to protect the content of the torrent. This link encryption would transform messages as they enter one hop and are forwarded to the next hop along multiple hop chains such that it is more difficult to link them.

7.2 Traffic Analysis Countermeasures

As previously described, an adversary may attempt to gain information about the peers through traffic analysis techniques. A group of colluding peers could monitor the piece requests and look for anomalous requests, such as multiple piece requests from the same peer. Also, malicious relay peers may issue requests for pieces and determine if those pieces are subsequently requested from another colluding peer. Using a timing correlation attack, it could be possible to identify some of the relay peers by using information related to the timing of the requests. A possible solution to this attack would be introduce intentional random delays as piece requests are forwarded. However, this would have a negative impact upon the system's performance.

Another defense against traffic analysis is to note that these colluding peers will typically have a limited number of IP addresses and will exhibit behaviors that deviate from standard peers and relay peers. If peers could share information in a privacy preserving manner, then they may be able to detect peers performing traffic analysis attacks and blacklist them from the torrent. We encourage future work aimed at addressing traffic analysis attacks in BitBlender and other privacy enhancing systems.

7.3 Selective Caching Policies

One technique that may mitigate an adversary's ability to conduct traffic analysis and simultaneously improve performance is the use of selective caching. As relay peers proxy requests, they could cache pieces in main memory as they are forwarded to the requester. As a consequence, the next time that a request is received for a piece that is cached, the relay peer could directly reply with the piece, rather than making another redundant request. This will reduce the expected path lengths for requests of pieces that reside in the relay's cache. Additionally, traffic analysis attempts may be frustrated, since the relay peers now behave as if they possess the requested piece. Using a selective caching policy, the expected path length presented in Section 5 becomes an upper bound. However, introducing a selective caching mechanism within an anonymous network exposes a variety of legal questions. In the next section, we provide a brief discussion of the potential legal liabilities that BitBlender (or *any* currently deployed anonymous network) presents.

8. LEGAL ISSUES

The success of BitBlender, or of any anonymous network, is dependent on the legality of operating an open relay. Anonymous networks can be used to enhance online privacy, enable free speech, and protect human rights; however, they can also be used to hide the identities of people engaged in illegal activities. Operators of relay nodes in networks such as Tor are sometimes accused of preforming illegal activities, since they appear to originate at the relay node [15]. Even though operators of these relay nodes are not directly causing harm, most Western countries have the legal notion of *indirect liability*, where one party can be held responsible for the actions of another party. There are two common categories of indirect liability – *vicarious liability* and *contributory liability*.

Vicarious liability arises when a third party has the ability, duty, or right to control the actions of another party. The main factors for a third party to be held vicariously liable are that they either enable or benefit from these actions. For example, if a bartender serves alcohol to a minor, the bar owner can be held liable for the actions of the bartender. Another common example is that parents or legal guardians can be liable for the wrong-doings of minors. However, phone companies are normally not liable for prank calls placed by their customers. It is unclear if the protection granted to phone companies also applies to Internet Service Providers (ISPs). It is equally unclear if operators of relay nodes in anonymizing multi-hop networks can be held vicariously liable for the actions of other users. However, United States law has a provision that makes caching and retransmission of unmodified cached files legal [1]. The law was upheld when Google's caching policy was challenged [4].

Contributory liability, occurs when a third party has induced or has reasonable knowledge of wrong-doing and fails to act to prevent these actions. An example of when a third party can be found contributorily liable is if an ISP receives notice that there is copyright infringing material present on their network and does not remove the infringing material. It is unclear what responsibility operators of relay nodes have for removing copyright infringing material that is transmitted through their nodes. It is clear from the United States Supreme Court ruling against Grokster [5] that the operators of an anonymous network are more likely to be found liable for inducing illegal activity if they advertise their tools as mechanisms to commit infringement, although many other factors are also important. This means that it is essential that anonymizing multi-hop networks be advertised as tools to enhance privacy and enable anonymous speech.

9. RELATED WORK

There have been numerous privacy enhancing systems proposed, the majority of which are based upon *mix networks* [7] or *onion routing* [13]. One of the first low latency anonymous networks was Crowds [19], which uses a loose routing scheme that constructs paths between the initiator and the destination server of non-deterministic lengths. Crowds provides anonymity for web transactions by forwarding each message to an intermediate "jondo" node or delivering it to the final destination server with a certain probability. To protect against local eavesdroppers, all communication between jondos is encrypted. BitBlender can be regarded as an incarnation of Crowds that is optimized particularly for BitTorrent traffic.

Tarzan [12] provides a peer-to-peer anonymous network layer with a high degree of resistance to traffic analysis by employing cover traffic in addition to using a source-routed mix strategy. Its peer membership list is maintained using a distributed hash table (DHT).

Anonymous networks with the ability to provide varying degrees of anonymity by tuning system parameters have also been proposed. $\mathcal{P}^5$ [22] uses broadcast channels and groups to allow users to choose between performance and anonymity.

von Ahn *et al.* proposed efficient anonymity protocols that provide k-anonymity for both the sender and receiver [25]. They present the argument that while k-anonymity is a weaker form of anonymity in comparison to that which is provided by mix or onion routing networks, it is sufficient in many cases, depending on the application and the sensitivity of the communications.

Tor [11] has become the most popular tool for providing anonymity to TCP-based applications. Tor's success is largely a result of its ability to provide strong anonymity properties in addition to a low latency transport service that is sufficient for anonymizing interactive applications such as web browsing and instant messaging. Tor utilizes centralized directory servers to maintain and distribute the list of active routers, which is used to construct three-hop circuits that are secured with a layered encryption technique based on onion routing [13].

BitBlender differs from these privacy enhancing systems in that it is specifically designed to efficiently anonymize BitTorrent traffic. As a result, it achieves its anonymity properties with very minimal protocol overhead. BitBlender is most similar to Crowds; however, it has the advantage that it does not need to employ any data confidentiality mechanism to protect against local eavesdroppers since all torrent data is publicly available.

10. CONCLUSION AND FUTURE WORK

We present BitBlender, an efficient protocol that aims to offer a usable and inter-operable anonymity layer for BitTorrent. In contrast to several existing privacy enhancing systems that provide anonymity for general-purpose traffic, we explore the design of a protocol-specific service that does not rely upon cryptography to achieve its anonymity properties. We show that such a design offers increased performance and adequate anonymity properties for the purpose of file transfers.

BitBlender builds an ad-hoc relay network that offers plausible deniability for the initiators of piece requests. We argue that this degree of anonymity is sufficient to obscure the identities of peers participating in a file download through BitTorrent. In addition, the protocol has the ability to dynamically adjust the degree of anonymity provided for the torrent based upon the adjustment of system parameters, specifically the number of relay peers present in an anonymous torrent.

As future work, we propose studies aimed at exploring the feasibility of confidentiality and access control mechanisms within this framework. Also, additional work is necessary to adequately study traffic analysis attacks and provide practical solutions. Finally, performance improvements through the use of various caching policies should be explored further. Systems that offer a level of anonymity that is appropriate for the degree of anonymity required are an intriguing concept and deserve additional research.

11. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their comments and suggestions. We also thank Paul Ohm for a helpful discussion of the legal issues. This work was partially funded by the National Science Foundation under grant ITR-0430593 and by IBM under a Faculty Partnership Award. D. McCoy was supported in part by gifts from his father Keith Martin.

12. REFERENCES

[1] 17 United States Code Section 512. http://www4.law.cornell.edu/uscode/17/512.html.

[2] Azureus BitTorrent client. `http://azureus.sourceforge.net`.

[3] Enhanced CTorrent. `http://www.rahul.net/dholmes/ctorrent`.

[4] Field v. Google, Inc., 412 F. Supp 2d. 1106 (D. Nev. 2006).

[5] MGM Studios Inc. v. Grokster, ltd., 545 U.S. 913 (Supreme Court 2005).

[6] A. Acquisti and R. Gross. Imagined communities: Awareness, information sharing, and privacy on the facebook. In *Privacy Enhancing Technologies*, pages 36–58, 2006.

[7] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.

[8] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 46–66, July 2000.

[9] R. Clayton, G. Danezis, and M. G. Kuhn. Real world patterns of failure in anonymity systems. In I. S. Moskowitz, editor, *Proceedings of Information Hiding Workshop (IH 2001)*, pages 230–244. Springer-Verlag, LNCS 2137, April 2001.

[10] C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.

[11] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[12] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, DC, November 2002.

[13] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. Hiding routing information. In R. Anderson, editor, *Proceedings of Information Hiding: First International Workshop*, pages 137–150. Springer-Verlag, LNCS 1174, May 1996.

[14] L. Lamport, R. E. Shostak, and M. C. Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[15] D. McCoy, K. Bauer, D. Grunwald, T. Kohno, and D. Sicker. Shining light in dark places: Understanding the Tor network. In *Proceedings of the 8th Privacy Enhancing Technologies Symposium*, July 2008.

[16] L. Peterson, S. Muir, T. Roscoe, and A. Klingaman. Planetlab architecture: An overview. Technical Report PDN–06–031, PlanetLab Consortium, May 2006.

[17] A. Pfitzmann and M. Hansen. Anonymity, unobservability, and pseudonymity: A consolidated proposal for terminology. Draft, July 2000.

[18] M. Piatek, T. Kohno, and A. 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.

[19] M. Reiter and A. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), June 1998.

[20] M. Rennhard and B. Plattner. Introducing MorphMix: Peer-to-peer based anonymous Internet usage with collusion detection. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2002)*, Washington, DC, USA, November 2002.

[21] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.

[22] R. Sherwood, B. Bhattacharjee, and A. Srinivasan. P5: A protocol for scalable anonymous communication. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, May 2002.

[23] A. Singh, P. Druschel, and D. S. Wallach. Eclipse attacks on overlay networks: Threats and defenses. In *IEEE INFOCOM*, 2006.

[24] L. Sweeney. k-Anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[25] L. von Ahn, A. Bortz, and N. J. Hopper. k-Anonymous message transmission. In V. Atluri and P. Liu, editors, *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS 2003)*, pages 122–130. ACM Press, October 2003.

[26] M. Wright, M. Adler, B. N. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. In *Proceedings of the Network and Distributed Security Symposium - NDSS '02*. IEEE, February 2002.

[27] M. K. Wright, M. Adler, B. N. Levine, and C. Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Trans. Inf. Syst. Secur.*, 7(4):489–522, 2004.