Internet Research Task Force (IRTF)
Request for Comments: 5765
Category: Informational
ISSN: 2070-1721

H. Schulzrinne
Columbia University
E. Marocco
Telecom Italia
E. Ivov
SIP Communicator

February 2010

Security Issues and Solutions in Peer-to-Peer Systems for Realtime Communications

Abstract

Peer-to-peer (P2P) networks have become popular for certain applications and deployments for a variety of reasons, including fault tolerance, economics, and legal issues. It has therefore become reasonable for resource consuming and typically centralized applications like Voice over IP (VoIP) and, in general, realtime communication to adapt and exploit the benefits of P2P. Such a migration needs to address a new set of P2P-specific security problems. This document describes some of the known issues found in common P2P networks, analyzing the relevance of such issues and the applicability of existing solutions when using P2P architectures for realtime communication. This document is a product of the P2P Research Group.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Research Task Force (IRTF). The IRTF publishes the results of Internet-related research and development activities. These results might not be suitable for deployment. This RFC represents the consensus of the Peer-to-Peer Research Group of the Internet Research Task Force (IRTF). Documents approved for publication by the IRSG are not a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5765.

Table of Contents

1. Introduction

Peer-to-peer (P2P) overlays have become quite popular with the advent of file-sharing applications such as Napster [NAPSTER], KaZaa [KAZAA], and BitTorrent [BITTORRENT]. After their success in file-sharing and content distribution [Androutsellis-Theotokis], P2P networks are now also being used for applications such as Voice over IP (VoIP) [SKYPE] [Singh] and television [PPLIVE] [COOLSTREAM]. However, most of these systems are not purely P2P and have centralized components like the login server in Skype [Baset] or moderators and trackers in BitTorrent [Pouwelse]. Securing pure P2P networks is therefore still a field of very active research [Wallach].

P2P overlays can be broadly classified as structured and unstructured [RFC4981], depending on their routing model. Unstructured overlays are often relatively simple, but search operations in them, usually based on flooding, tend to be inefficient. Structured P2P overlays use distributed hash tables (DHTs) [Stoica] [Maymounkov] [Rowstron] to perform directed searches, which make lookups more efficient in locating data. This document will mostly focus on DHT-based P2P overlays.

When analyzing the various attacks that are possible on P2P systems, it is important to first understand the motivation of the attackers as well as the resources (e.g., computation power, access to different IP subnets) that they would have at their disposal.

Once the threat has been identified, admission control is a first step towards security that can help avoid a substantial number of attacks [Kim]. Most solutions rely on the assumption that malicious nodes represent a small fraction of all peers. It is therefore important to restrict their number in the overlay.

Other P2P-specific security problems discussed here include attacks on the routing of queries, targeted denial-of-service attacks, and attacks on data integrity.

In the remainder of this document, we outline the main security issues and proposed solutions for P2P systems. Following this, we focus on a particular class of P2P applications that provide realtime communications. Realtime communications use the same DHTs used by file-sharing applications; however, the data that is saved in these DHTs is different. In realtime communications, the contents stored in the DHTs comprises user location, the DHT being the substitute for a centralized registration server.

At first glance, it may appear that requirements on peer-to-peer systems for realtime communication services are no different than those for file-sharing services. Table 1 demonstrates that there are sizeable differences related to privacy, availability, and a marked increase in the general security requirements.

	File-sharing	Realtime communication
Distributed database	Shared file locations are indexed in a table distributed among peers; often hundreds or thousands per peer.	User locations are indexed in a table distributed among peers; rarely more than one per peer.
Availability	Same files are usually available at multiple locations and failures involving single instances are overcome by abundancy of resources; attacks targeting single files need to be addressed to the distributed index.	Users are unique; attacks targeting single users may be addressed both to the distributed index and to the user's device directly.
Integrity	Attackers may want to share corrupted files in place of popular content, e.g., to discourage users from acquiring copyrighted material; constitute a threat for the service, but not for the users.	Attackers may want to impersonate different users in order to handle calls directed to them; constitute a particular threat for the user as, in case of success, the attacker acquires full control on the victim's personal communications.
Confidentiality	Shared files are, by definition, readable by all users; in some cases, encryption is used to avoid elements not involved in the service to detect traffic.	Communications are usually meant to be private and need to be encrypted; eavesdropping may reveal sensitive data and is a serious threat for users.

Bitrate and latency	The file-transfer use case is particularly tolerant to unstable bitrates and ability to burst on and off as peers disappear or new ones become available.	Realtime traffic almost always requires a constant minimum bitrate and low latency in order to avoid problems like jitter. While this is not directly related to a specific sort of attacks, it is a significant constraint to the design of certain design solutions, and in particular those that somehow affect routing.
Peer lifetime	File-sharing users do not need to stay in the overlay more than the time required for downloading the content they are looking for.	Realtime communication applications need not leave the overlay for as long as the user wants to stay connected and be reachable. This gives the attackers longer time for conducting successful targeted attacks.

Table 1: Main differences between P2P applications used for file-sharing and for realtime communication.

1.1. Purpose of This Document

The goal of this document is to provide authors of P2P protocols for realtime communications with background that they may find useful while designing security mechanisms for specific cases. The document has been extensively discussed during face-to-face meetings and on the P2PRG mailing list; it has been reviewed both substantially and editorially by two members of the research group and reflects the consensus of the group.

The content of this document was partially derived from the article "Peer-to-peer Overlays for Real-Time Communication: Security Issues and Solutions," published in IEEE Surveys & Tutorials, Vol. 11, No. 1, and originally authored by Dhruv Chopra, Henning Schulzrinne, Enrico Marocco, and Emil Ivov.

It is important to note that this document considers "security" from the perspective of application developers and protocol architects. It is hence entirely agnostic to potential legislation issues that may apply when protecting applications against a specific attack, as, for example, in the case of lawful interception.

1.2. Structure of This Document

The document is organized as follows. In Section 2, we discuss P2P security attackers. We try to elaborate on their motivation, the resources that would generally be available to them, their victims, and the timing of their attacks. In Section 3, we discuss admission control problems. In Section 4, we identify the problem of where a node joins in the overlay. In Section 5, we describe problems related to identification of malicious nodes and the dissemination of this information. In Section 6, we describe the issues of routing and data integrity in P2P networks. Finally, in Section 7 we discuss how issues and solutions previously presented apply in P2P overlays for realtime communication.

Table 2 and Table 3 provide an index of the attacks and the solutions discussed in the rest of this document.

Attack name	Referring sections
botnets (use of)	Section 2.1, Section 2.2
denial of service (DoS)	Section 2.1, Section 7.2.1
man in the middle (MITM)	Section 7.2.2
poisoning	Section 6.1, Section 7.2.2
pollution	Section 2.1, Section 6.1
sybil	Section 2.2, Section 4
targeted denial of service	Section 7.2.1

Table 2: Index of some of the more popular attacks and problems discussed in this document.

Solution name	Referring sections
admission control	Section 3
anonymity	Section 5.2
asymmetric key pair	Section 7.2.5
CAPTCHA	Section 3
certificates	Section 7.2.3
CONNECT (SIP method)	Section 7.2.4
cryptographic puzzles	Section 4
diametrically opposite IDs	Section 4
end-to-end encryption	Section 7.2.4
group authority	Section 3
group charter	Section 3
iterative routing	Section 7.2.2
no profit for newcomers	Section 5.2
online phone book	Section 7.2.5
passive upgrades	Section 7.1.1
peer promotion	Section 7.1
proactive identification	Section 5.1.1
reactive identification	Section 5.1.2
recommendation	Section 3
reputation management systems	Section 5.2
self-policing	Section 5.2
signatures	Section 3
social networks (using)	Section 4, Section 6.2,
SRTP	Section 7.2.6
structured reputation management	Section 5.2.2
SybilGuard (protocol)	Section 4
transitivity of trust	Section 5.2.2
trust and distrust vectors	Section 5.2.1
trust and trusted nodes	Section 3, Section 6.2, Section 7.2.3
unstructured reputation management	Section 5.2.1
voluntary moderators	Section 6.1

Table 3: Index of some of the more popular solutions discussed in this document.

2. The Attackers

2.1. Incentive of the Attacker

Attacks on networks happen for a variety of reasons such as monetary gain, personal enmity, or even for fame in the hacker community.

There are quite a few well-known cases of denial-of-service attacks for extortion in the client-server model [McCue]. One of the salient points of the P2P model is that the services it provides have higher robustness against failure. However, denial-of-service attacks are still possible against individuals within the overlay if the attackers possess sufficient resources. For instance, a network of worm-infected malicious nodes spread across the Internet and controlled by an attacker (often referred to as botnet) could simultaneously bombard lookup queries for a particular key in the DHT. The peer responsible for this key would then come under a lot of load and could crash [Sit]. However, with replication of key-value pairs at multiple locations, such threats can be mitigated.

Attackers may also have other incentives indirectly related to money. With the growth of illegal usage of sharing files with copyrights, record companies have been known to pollute content in the overlays by putting up nodes with corrupt chunks of data but with correct file names to degrade the service [Liang] and in hope that users would get frustrated and stop using it. Similarly, competition between different communication service providers, either or both based on P2P technologies, and the low level of traceability of attacks targeted to single users could be considered as motivation for attempting service disruption.

Attacks can also be launched by novice attackers who are attacking the overlay for fun or fame in a community. These are perhaps less likely to be successful or cause damage, since their resources tend to be relatively limited.

2.2. Resources Available to the Attacker

Resource constraints play an important role in determining the nature of the attack. An attacker who controls a botnet can use an Internet relay channel and launch distributed denial-of-service attacks against another node. With respect to attacks where a single node impersonates multiple identities, as in the case of the Sybil attack [Douceur] described in Section 4, IP addresses are also an important resource for the attacker since in DHTs such as Chord [Stoica], the position in the overlay is determined by using a base hash function such as SHA-1 [SHA1] on the node's IP address. The cryptographic puzzles [Rowaihy] that are sometimes suggested as a way to deter Sybil attacks by making the join process harder are futile against an attacker with a botnet and virtually unlimited computation power. Douceur [Douceur] proves that even with the assumption that attackers only have minimum resources at their disposal, it is not possible to defend against them in a pure P2P system.

2.3. Victim of the Attack

The victim of an attack could be an individual node, a particular content entry, or the entire overlay service. If malicious nodes are strategically placed in the overlay, they can block a node from using its services. Attacks could also be launched against specific content [Sit] or even the entire overlay service. For example, if the malicious nodes are randomly placed in the overlay and drop packets or upload malicious content, then the quality of the overlay would deteriorate.

2.4. Time of Attack

A malicious node could start misbehaving as soon as it enters the overlay or it could follow the rules of the overlay for a finite amount of time and then attack. The latter could prove to be more harmful if the overlay design suggests accumulating trust in peers based on the amount of time they have been present and/or not misbehaving. In Kademlia [Maymounkov], for instance, the routing tables are populated with nodes that have been up for a certain amount of time. While this provides some robustness from attacks in which the malicious nodes start dropping routing requests from the moment they enter, it would take time for the algorithm to adapt to nodes that start misbehaving in a later stage (i.e., after they have been recorded in routing tables). Similarly for reputation management systems, it is important that they adapt to the current behavior of a peer.

3. Admission Control

Admission control depends on who decides whether or not to admit a node and how this permission is granted. Kim et al. [Kim] answer these questions independently of any particular environment or application. They define two basic elements for admission in a peer group, a group charter, which is an electronic document that specifies the procedure of admission into the overlay, and a group authority, which is an entity that can certify group admission. A prospective member first gets a copy of the group charter, satisfies the requirements, and approaches the group authority. The group authority then verifies the admission request and grants a group membership certificate.

The group charter and authority verification can be provided by a centralized certificate authority or a trusted third party, or it could be provided by the peers themselves (by voting). The former is more practical and tends to make the certification process simpler although it is in violation of the pure P2P model and exposes the system to attacks typical for server-based solutions (e.g., denial-

of-service attacks targeted to the central authority). In the latter case, the group authority could either be a fixed number of peers or it could be a dynamic number based on the total membership of the group. The authors argue that even if the group charter requires a prospective member to get votes from peers, the group membership certificate must be issued by a distinct entity. The reason for this is that voters need to accompany their votes with a certificate that proves their own membership. Possible signature schemes that could be used in voting such as plain digital signature, threshold signature, and accountable subgroup multisignature are also described. Saxena et al. [Saxena] performed experiments with the different signature schemes and suggest the use of plain signatures for groups of moderate size and where bandwidth is not a concern. For larger groups and where bandwidth is a concern, they suggest threshold signature [Kong] and multisignature schemes [Ohta].

Another way of handling admission would be to use mechanisms based on trust and recommendation where each new applicant has to be known and vouched for by at least N existing members. The difficulties that such models represent include identity assertion and preventing bot/worm attacks. A compromised node could have a valid certificate identifying a trustworthy peer, and it would be difficult to detect this. Possible solutions include sending graphic or logic puzzles easily addressed by humans but hard to solve by computers, also known as CAPTCHA [Ahn]; however, reliability of such mechanisms is at the time of writing a topic of lively debate [Tam] [Chellapilla].

4. Determining the Position in the Overlay

For ring-based DHT overlays such as Chord [Stoica], Kademlia [Maymounkov], and Pastry [Rowstron], when a node joins the overlay, it uses a numeric identifier (ID) to determine its position in the ring. The positioning of a node determines what information it stores and which nodes it serves. To provide a degree of robustness, content and services are often replicated across multiple nodes. However, it is possible for an adversary with sufficient resources to undermine the redundancy deployed in the overlay by representing multiple identities. Such an attack is called a Sybil attack [Douceur]. This makes the assignment of IDs very important. One possible scheme to tackle such attacks on the ID mapping is to have a temporal mechanism in which nodes need to re-join the network after some time [Condie] [Scheideler]. Such temporal solutions, however, have the drawback that they increase the maintenance traffic and possibly deteriorate the efficiency of caching. Danezis et al. [Danezis] suggest mechanisms to mitigate the effect of Sybil attacks by reducing the amount of information received from malicious nodes. Their idea is to vary the nodes used for routing with time. This helps avoiding trust bottlenecks that may occur when applications

only route traffic through a limited set of highly trusted nodes. Other solutions suggest making the joining process harder by introducing cryptographic puzzles as suggested by Rowaihy et al. [Rowaihy]. The assumption is that the adversary has limited computational resources, which may not be true if the adversary has control over a botnet. Another drawback of such methods is that non-malicious nodes would also have to perform the extra computations before they can join the overlay.

A possible heuristic to hamper Sybil attacks is to employ redundancy at nodes with diametrically opposite IDs (in the DHT ID space) instead of successive IDs as in Chord. The idea behind choosing diametrically opposite nodes is based on the fact that a malicious peer can grant admission to others as its successor without them actually possessing the required IP address (whose hash is adjacent to the former's), and then they can cooperate to control access to that part of the ring. If, however, admission decisions and redundant content (for robustness) also involve nodes that are the farthest away (diametrically opposite) from a given position, then the adversary would require double resources (IP addresses) to attack. This happens because the adversary would need presence in the overlay at two independent positions in the ring.

Another approach proposed by Yu et al. [Yu] to limit Sybil attacks is based on the usage of the social relations between users. The solution exploits the fact that as a result of Sybil attacks, affected P2P overlays end up containing a large set of Sybil nodes connected to the rest of the peers through an irregularly small number of edges. The SybilGuard protocol [Yu] defines a method that allows to discover such kinds of discontinuities in the topology by using a special kind of a verifiable random walk and hence without the need of one node having a global vision of the graph.

It is also worth mentioning that in DHT overlays using different geometric concepts (e.g., hypercubes instead of rings), peer positions are usually not related to identifiers. In the content addressable network (CAN) [Ratnasamy], for example, the position of an entering node may be either selected by the node itself or, with little modification to the original algorithm, assigned by peers already in the overlay. However, even when malicious nodes do not know their position before joining, the overlay is still vulnerable to Sybil attacks.

5. Resilience against Malicious Peers

Making overlays robust against even a small percentage of malicious nodes is difficult [Castro]. It is therefore important for other peers to identify such nodes and keep track of their number. There

are two aspects to this problem. One is the identification itself, and the second is the dissemination of this information amongst the peers. Different metrics need to be defined depending on the peer group for the former, and reputation management systems are needed for the latter.

5.1. Identification of Malicious Peers

For identifying a node as malicious, malicious activity has to be observed first. This could be done in either a proactive way or a reactive way.

5.1.1. Proactive Identification

When acting proactively, peers perform periodic operations with the purpose of detecting malicious activity. A malicious node could prevent access to content for which it is responsible (e.g., by claiming the object doesn't exist), or return references to content that does not match the original queries [Sit]. With this approach, publishers of content can later perform lookups for it at periodic intervals and verify the integrity of whatever is returned. Any inconsistencies could then be interpreted as malicious activity. The problem with proactive identification is the management of the overhead it implies: if checks are performed too often, they may actually hinder scalability, while, if they are performed too rarely, they would probably be useless.

An additional approach for mitigating routing attacks and identifying malicious peers consists in sending multiple copies of the same message on different paths. With such an approach, implemented, for example, in Kademlia [Maymounkov], the sending peer can identify anomalies comparing responses coming in from different paths.

5.1.2. Reactive Identification

In a reactive strategy, the peers perform normal operations and if they happen to detect some malicious activity, then they can label the responsible node as malicious and avoid sending any further message to it. In a file-sharing application, for example, after downloading content from a node, if the peer observes that data does not match its original query it can identify the corresponding node as malicious. Poon et al. [Poon] suggest a strategy based on the forwarding of queries. If routing is done in an iterative way, then dropping of packets, forwarding to an incorrect node, and delay in forwarding arouse suspicion and the corresponding peer is identified as malicious.

5.2. Reputation Management Systems

Reputation management systems are used to allow peers to share information about other peers based on their own experience and thus help in making better judgments. Most reputation management systems proposed in the literature for file-sharing applications [Uzun] [Damiani] [Lee] [Kamvar] aim at preventing misbehaving peers with low reputation to rejoin the network with a different ID and therefore start from a clean slate. To achieve this, Lee et al. [Lee] store not only the reputation of a peer but also the reputation of files based on file name and content to avoid spreading of a bad file. Another method is to make the reputation of a new peer the minimum possible. Kamvar et al. [Kamvar] define five design considerations for reputation management systems:

- The system should be self-policing.
- The system should maintain anonymity.
- The system should not assign any profit to newcomers.
- The system should have minimal overhead in terms of computation, infrastructure, storage, and message complexity.
- The system should be robust to malicious collectives of peers who know one another and attempt to collectively subvert the system.

5.2.1. Unstructured Reputation Management

Unstructured reputation management systems have been proposed by Uzun et al. [Uzun] and Damiani et al. [Damiani]. The basic idea of these is that each peer maintains information about its own experience with other peers and resources, and shares it with others on demand. In the system proposed by Uzun et al. [Uzun], each node maintains trust and distrust vectors for every other node with which it has interacted. When reputation information about a peer is required, a node first checks its local database, and if insufficient information is present, it sends a query to its neighbors just as it would when looking up content. However, such an approach requires peers to get reputation information from as many sources as possible; otherwise, malicious nodes may successfully place targeted attacks returning false values for their victims.

5.2.2. Structured Reputation Management

One of the problems with unstructured reputation management systems is that they either take the feedback from few peers or, if they do so from all, then they incur large traffic overhead. Systems such as

those proposed by [Lee] [Kamvar] try to resolve it in a structured manner. The idea of the eigen trust algorithm [Kamvar], for example, is transitivity of trust. If a node trusts peer X, then it would also trust the feedback it gives about other peers. A node builds such information in an iterative way; for maintaining it in a structured way, the authors propose to use a content addressable network (CAN) DHT [Ratnasamy]. The information about each peer is stored and replicated on different peers to provide robustness against malicious nodes. They also suggest favoring peers probabilistically with high trust values instead of doing it deterministically, to allow new peers to slowly develop a reputation. Eventually, they suggest the use of incentives for peers with high reputation values.

6. Routing and Data Integrity

Preserving integrity of routing and data, or, in other words, preventing peers from returning corrupt responses to queries and routing through malicious peers, is an important security issue in P2P networks. The data stored on a P2P overlay depends on the applications that are using it. For file-sharing, this data would be the files themselves, their location, and owner information. For realtime communication, this would include user location bindings and other routing information. We describe such data integrity issues in Section 7.

6.1. Data Integrity

For file-sharing applications, insertion of wrong content (e.g., files not matching their names or descriptions) and introduction of corrupt data chunks (often referred to as poisoning and pollution) are a significant problem. BitTorrent uses voluntary moderators to weed out bogus files and the SHA-1 algorithm to determine the hash of each piece of a file to allow verification of integrity. If a peer detects a bad chunk, it can download that chunk from another peer. With this strategy, different peers download different pieces of a file before the original peer disappears from the network. However, if a malicious peer modifies the pieces that are only available on it and the original peer disappears, then the object distribution will fail [Zhang]. An analysis of BitTorrent in terms of integrity and performance can be found in the work of Pouwelse et al. [Pouwelse].

6.2. Routing Integrity

To enhance the integrity of routing, it is important to reduce the number of queries forwarded to malicious nodes. Marti et al. [Marti] developed a system that uses social network information to route queries over trusted nodes. Their algorithm uses trusted nodes

to forward queries (if one exists and is closer to the required ID in the ID space). Otherwise, they use the regular Chord [Stoica] routing table to forward queries. While their results indicate good average performance, it cannot guarantee log(N) hops for all cases. Danezis et al. [Danezis] suggest a method for routing in the presence of a large number of Sybil nodes. Their method is to ensure that a peer queries a diverse set of nodes and does not place too much trust in a node. Both the above works have been described based on Chord. However, unlike Chord, in DHTs like Pastry [Rowstron] and Kademlia [Maymounkov] there is flexibility in selecting nodes for any row in a peer's routing table. Potentially many nodes have a common ID prefix of a given length and are candidates for routing a given query. To exploit the social network information and still guarantee log(N) hops, a peer should select its friends to route a query, but only when they are present in the appropriate row selected by the DHT algorithm.

7. Peer-to-Peer in Realtime Communication

The idea of using P2P in realtime communication essentially implies distributing centralized entities from conventional architectures over P2P overlays and thus reducing the costs of deployment and increasing reliability of the different services. Initiatives such as the P2PSIP working group in IETF [P2PSIP] are currently concentrating on achieving this by using a DHT for services such as registration, location lookup, and support for NAT traversal, which are normally handled by dedicated servers.

Even if based on the same technology, overlays used for realtime communication differ from those used for file-sharing in at least two aspects:

- Resource consumption. Contrary to file-sharing systems where the DHT is used to store huge amounts of data (even if the distributed database is used only for storing file locations, each user usually indexes hundreds or thousands of files), realtime communication overlays only require a subset of the resources available at any given time as users only register a limited number of locations (rarely more than one).

- Confidentiality. In file-sharing applications, eavesdropping and identity theft do not constitute real threats; after all, files are supposed to be made publicly available. This is not true in realtime communications, where the privacy and confidentiality of the participants are of paramount importance. Furthermore, the notion of identity plays an important role in realtime

communications since it is the basis for starting a communication session. As such, it is essential to have mechanisms to unequivocally assert identities in realtime communication systems.

In this section we go over the admission issues and security problems discussed in previous sections, and discuss solutions that would be applicable to realtime communication in P2P.

7.1. Peer Promotion

In order to remain compatible with existing user agents, P2P communication architectures would have to allow certain nodes to use their services without actually using overlay-specific semantics. One way to achieve this would be for overlay-agnostic nodes to register with an existing peer or a dedicated proxy via a standard protocol like SIP [RFC3261]. Through the rest of this document, we will refer to nodes that access the service without actually joining the overlay as "clients".

In most cases, users would be able to benefit from the overlay by only acting as clients. However, in order to keep the solution scalable, at some point clients would have to be promoted to peers (admission to the DHT). This requires addressing the following issues.

7.1.1. Active vs. Passive Upgrades

Most existing P2P networks [KAZAA] [BITTORRENT] [PPLIVE] would generally leave it to the clients to determine if and when they would apply for becoming peers. A well-known exception to this trend is the Skype network [SKYPE], arguably one of the most popular overlay networks used for realtime communications today. Instances of the Skype application are supposed to operate as either super-nodes, directly contributing to the distributed provision of the service, or ordinary-nodes, simply using the service, and the "promotions" are decided by the higher levels of the hierarchy [Baset]. Even if there is not much difference for a client whether it has to actively ask for authorization to join an overlay or passively wait for an invitation, the latter approach has some advantages that fit well in overlays where only a subset of the peers is required to provide the service (as in realtime communication):

- An attacker cannot estimate in advance when and if it would be invited to join the overlay as a peer.

- It allows peers to perform long-lasting measurements on sets of candidates, in order to accurately select the most appropriate for upgrading and only invite it when they are "ready" to do so. The

opposite approach, that is, when clients initiate the join themselves, adds an extra constraint for the peer that has to act upon the request since it doesn't know if and when the peer would attempt to join again.

- It discourages malicious peers from attempting Sybil and, more generally, brute force attacks, as only a small ratio of clients has chances to join the overlay (possibly after an accurate examination).

7.1.2. When to Upgrade

In order to answer this question, one would have to define some criteria that would allow determination of the load on a peer and a reasonable threshold. When the load exceeds this threshold, a client is invited to become a peer and share the load. Several mechanisms to diagnose the status of P2P systems have recently been proposed [P2PSIP-DIAG]; in general, reasonable criteria for determining load can be:

- Number of clients attached.
- Bandwidth usage for DHT maintenance, forwarding requests, and responses to and from peers and from the attached clients.
- Memory usage for DHT routing table, DHT neighborhood table, application-specific data, and information about the attached clients.

7.1.3. Which Clients to Upgrade

Selecting which clients to upgrade would require defining and keeping track of new metrics. The exact set of metrics and how they influence decisions should be the subject of serious analysis and experimentation. These could be based on the following observations:

- Uptime. A peer could easily record the amount of time that it has been maintaining a connection with a client and take it into account when trying to determine whether or not to upgrade it.
- Level of activity. It is reasonable to assume that the more a client uses the service (e.g., making phone calls), the less they would be willing to degrade it.
- Keeping track of history. Peers could record history of the clients they invite and the way they contribute to the overlay.

Other metrics such as public vs. private IP addresses, computation power, and bandwidth should also be taken into account even though they do not necessarily have a direct impact on security.

Note however that a set of colluded malicious peers can manufacture basically any criteria considered for the upgrade. Furthermore, sophisticated peers can overload the system or run denial-of-service attacks against existing super-nodes in order to improve their chances of being upgraded.

7.1.4. Incentives for Clients

Clients need to have incentives for accepting upgrades in order to prevent excessive burden on existing peers. One way to handle this would be to maintain separate incentive management through the use of currency or credits. Another option would involve embedding these incentives inside the protocol itself:

- Peers share with clients only a fraction of their bandwidth (uplink and downlink). This would result in higher latency when using the services of the overlay as a client and better service quality for peers.

- Peers could restrict the number or types of calls that they allow clients to make.

Introducing such incentives, however, may turn out to be somewhat risky. Differences in quality would probably be perceptible for end users who would not always be able to understand the difference between the roles that their user agent is playing in the overlay. Such behavior may therefore be interpreted as arbitrary and make the service look unreliable.

7.2. Security

7.2.1. Targeted Denial of Service

In addition to bombardment with queries as described in Section 2, the denial-of-service attack against an individual node can be conducted in DHTs if the peers that surround a particular ID are compromised. These peers that act as proxy servers for the victim can fake the responses from the victim by sending fictitious error messages back to peers trying to establish a session. Danezis et al.'s solution [Danezis] can also provide protection against such attacks, as in their solution peers vary the nodes used in queries.

7.2.2. Man-in-the-Middle Attack

The man-in-the-middle attack is well described by Seedorf [Seedorf1] in the particular case of P2PSIP [P2PSIP] and consists of an attack that exploits the lack of integrity when routing information. A malicious node could return IP addresses of other malicious nodes when queried for a particular ID. The requesting peer would then establish a session with a second malicious node, which would again return a "poisoned" reply. This could go on until the Time to Live (TTL) expires and the requester gives up the "wild goose chase" [Danezis]. A simple way for entities to verify the correctness of the routing lookup is to employ iterative routing and to check the node-ID of every routing hop that is returned, and it should get closer to the desired ID with every hop. However, this is not a strong check and can be defeated [Seedorf1].

7.2.3. Trust between Peers

The effect of malicious peers could be mitigated by introducing the concept of trust within an overlay. This can be done in different ways:

- Using certificates assigned by an external authority. The drawback with this approach is that it requires a centralized element.

- Using certificates reciprocally signed by peers. This mechanism is quite similar to PGP [Zimmermann]; every peer signs certificates of "friend" peers and trusts any other peer with a certificate signed by one of its friends. However, even though it might be theoretically possible, in reality it is extremely difficult to obtain long enough trust chains.

7.2.4. Routing Call Signaling

One way for implementing realtime communication overlays (as we have mentioned in earlier sections) would be to simply replace centralized entities in signaling protocols like SIP [RFC3261] with distributed services. In some cases, this might imply reusing existing protocol mechanisms for routing signaling messages. In the case of SIP, this would imply regarding peers as SIP proxies. However, the design of SIP supposes that such proxies are trusted, and makes it possible for them to fork requests or change their destination, add or remove header fields, act as the remote party, and generally manipulate message content and semantics.

However, in a P2P environment where messages may be routed through numerous successive peers, some of which might be compromised, it is important not to treat them as trusted proxies. One way to limit what peers can do is by protecting signaling with some kind of end-to-end encryption.

Another option would be to extend existing signaling protocols and modify the way they route messages in order to guarantee secure end-to-end transmission. Gurbani et al. [Gurbani] define a similar mechanism for SIP that allows nodes to establish a secure channel by sending a CONNECT SIP request, and then tunnel all SIP messages through it, adopting a similar mechanism to the one used for upgrading from HTTP to HTTPS [RFC2818].

7.2.5. Integrity of Location Bindings

It is important to ensure that the location that a user registers, usually a (URI, IP) pair, is what is returned to the requesting party. Or the entities that issue the lookup request must be able to verify the integrity of this pair. A pure P2P approach to allow verification of the integrity of location binding information is presented in [Seedorf2]. The idea is for an entity to choose an asymmetric key pair and hash its public key to generate its URI. The entity then signs its present location with its private key and registers with the quadruple (URI, IP, signature, public key). Any entity that looks up the URI and receives such a quadruple can then verify its integrity by using the public key and the certificate. Another possible merit of such an approach could be that it is possible to identify the malicious nodes and maintain a black list. However, the resulting URIs are not easy to remember and associate with entities. Discovering these URIs and associating them with entities would therefore require some sort of a directory service. The authors suggest using existing authentication infrastructure for this such as a certified web service using SSL that can publish an "online phone book" mapping users to URIs.

7.2.6. Encrypting Content

Using P2P overlays for realtime communication implies that content is likely to traverse numerous intermediate peers before reaching its destination. A typical example could be the use of peers as media relays as a way of traversing NATs in VoIP calls.

Contrary to publicly shared files, communication sessions are in most cases expected to be private. It is therefore very important to make sure that no media leaves the client application without being encrypted and securely transported through a protocol like SRTP [RFC3711]. However, the processing required by the encryption

algorithms and the extra resources necessary for managing the keying material (e.g., for retrieving public keys when interacting with unknown peers) may be expensive, especially for mobile devices.

7.2.7. Other Issues

Details on cost and payment regimes could help identify further threats. Such details could also be important when determining the impact of a potential attack in the context of the specific business models associated with particular overlays. In many cases, answers to the following simple questions significantly aid the design of protection mechanisms:

- Whom do the users pay?
- Do the users only pay when accessing the public telephone network?
- Is the billing done per call or is it fixed?

For instance, the implications of an attack such as taking control over another's user agent or its identity and using it for outbound calls would depend on whether or not this would be economically advantageous for the attacker. Baumann et al. [Baumann] suggest that to prevent unwanted communication costs, gateways for the public telephone network should only be accessible via authenticated servers and dialing authorizations should be enforced. Also, it seems that it would be difficult to do billing in a pure P2P manner as it would mean keeping the billing details with untrusted peers.

8. Open Issues

Existing systems used for file-sharing, media streaming, and realtime communications all achieve a reasonable level of security relying on centralized components (e.g., login servers in Skype [Baset], moderators and trackers in BitTorrent [Pouwelse]). Securing pure P2P networks is therefore still a very active research field; at the time of writing the main open issues fall in five areas:

- Secure assignment of node IDs.
- Entity-identity association.
- Distributed trust among peers.
- Resistance against malicious peer collusion.
- Robustness and damage recovery.

In general, P2P overlays are designed to work when the vast majority of their peers are interested in the service provided by the system and act benevolently. Understanding how operations in different overlays are perturbed as the number of malicious or compromised peers grows is another interesting area of research. Also, a widely adopted methodology for the evaluation and classification of security solutions would be likely to help research in the field of P2P security progress more efficiently.

9. Security Considerations

This document, tutorial in nature, discusses some of the security issues of P2P systems used for realtime communications. It does not aim at identifying all possible threats and the corresponding solutions; instead, starting from an analysis of the attackers, it delves into some important aspects of P2P security, referencing the most relevant works published at the time of writing and discussing how they apply (or could apply) to the case of realtime communications.

10. Acknowledgments

The authors are particularly grateful to Dhruv Chopra, who contributed to the writing of the article "Peer-to-peer Overlays for Real-Time Communication: Security Issues and Solutions" (IEEE Surveys & Tutorials, Vol. 11, No. 1) from which this work is partially derived.

The authors would also like to thank Vijay Gurbani and Song Haibin for reviewing the document and the many others who provided useful comments.

11. Informative References

[Ahn] Ahn, L., Blum, M., and J. Langford, "Telling humans and computers apart automatically", Communications of the ACM, vol. 47, no. 2, February 2004.

[Androutsellis-Theotokis] Androutsellis-Theotokis, S. and D. Spinellis, "A survey of peer-to-peer content distribution technologies", ACM CSUR, vol. 36, no. 4, December 2004.

[BITTORRENT] "BitTorrent", <http://www.bittorrent.com/>.

[Baset] Baset, S. and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol", Proceedings of IEEE INFOCOM 2006, April 2006.

[Baumann] Baumann, R., Cavin, S., and S. Schmid, "Voice Over IP - Security and SPIT", Technical Report, University of Berne, September 2006.

[COOLSTREAM] "COOLSTREAMING", <http://www.coolstreaming.us>.

[Castro] Castro, M., Druschel, P., Ganesh, A., Rowstron, A., and D. Wallach, "Secure routing for structured peer-to-peer overlay networks", Proceedings of 5th symposium on Operating systems design and implementation, December 2002.

[Chellapilla] Chellapilla, K. and P. Simard, "Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)", Proceedings of Advances in Neural Information Processing Systems, December 2004.

[Condie] Condie, T., Kacholia, V., Sankararaman, S., Hellerstein, J., and P. Maniatis, "Maelstorm: Churn as Shelter", Proceedings of 13th Annual Network and Distributed System Security Symposium, November 2005.

[Damiani] Damiani, E., Vimercati, D., Paraboschi, S., Samarati, P., and F. Violante, "A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks", Proceedings of Conference on Computer and Communications Security, November 2002.

[Danezis] Danezis, G., Lesniewski-Laas, C., Kaashoek, M., and R. Anderson, "Sybil-resistant DHT routing", Proceedings of 10th European Symposium on Research in Computer Security, September 2005.

[Douceur] Douceur, J., "The Sybil Attack", Revised Papers from First International Workshop on Peer-to-Peer Systems, March 2002.

[Gurbani] Gurbani, V., Willis, D., and F. Audet, "Cryptographically Transparent Session Initiation Protocol (SIP) Proxies", Proceedings of IEEE ICC ’07, June 2007.

[KAZAA] "KaZaa", <http://www.kazaa.com/>.

[Kamvar] Kamvar, S., Garcia-Molina, H., and M. Schlosser, "The EigenTrust Algorithm for Reputation Management in P2P Networks", Proceedings of 12th international conference on World Wide Web, May 2003.

[Kim] Kim, Y., Mazzocchi, D., and G. Tsudik, "Admission Control in Peer Groups", Proceedings of Second IEEE International Symposium on Network Computing and Applications, April 2003.

[Kong] Kong, J., Zerfos, P., Luo, H., Lu, S., and L. Zhang, "Providing robust and ubiquitous security support for MANET", Proceedings of 9th International Conference on Network Protocols, November 2001.

[Lee] Lee, S., Kwon, O., Kim, J., and S. Hong, "A Reputation Management System in Structured Peer-to-Peer Networks", Proceedings of 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, June 2005.

[Liang] Liang, J., Kumar, R., Xi, Y., and K. Ross, "Pollution in p2p file sharing systems", Proceedings of IEEE INFOCOM 2005, March 2005.

[Marti] Marti, S., Ganesan, P., and H. Garcia-Molina, "SPROUT: P2P Routing with Social Networks", Proceedings of First International Workshop on Peer-to-Peer and Databases, March 2004.

[Maymounkov] Maymounkov, P. and D. Mazi, "Kademlia: A Peer-to-peer Information System Based on the XOR Metric", Proceedings of First International Workshop on Peer-to-peer Systems, March 2002.

[McCue] McCue, Andy., "Bookie reveals 100,000 cost of denial-of-service extortion attacks", available from http://www.silicon.com, June 2004.

[NAPSTER] "Napster", <http://www.napster.com/>.

[Ohta] Ohta, K., Micali, S., and L. Reyzin, "Accountable Subgroup Multisignatures", Proceedings of 8th ACM conference on Computer and Communications Security, November 2001.

[P2PSIP] "Peer-to-Peer Session Initiation Protocol (P2PSIP) IETF Working Group", <http://www.ietf.org/html.charters/p2psip-charter.html>.

[P2PSIP-DIAG] Yongchao, S., Jiang, X., Even, R., and D. Bryan, "P2PSIP Overlay Diagnostics", Work in Progress, December 2009.

[PPLIVE] "PPLive", <http://www.pplive.com>.

[Poon] Poon, W. and R. Chang, "Robust Forwarding in Structured Peer-to-Peer Overlay Networks", Proceedings of ACM SIGCOMM 2004, August 2004.

[Pouwelse] Pouwelse, J., Garbacki, P., Epema, D., and H. Sips, "The Bittorent P2P File-Sharing System: Measurements and Analysis", Proceedings of 4th International Workshop of Peer-to-peer Systems, February 2005.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4981] Risson, J. and T. Moors, "Survey of Research towards Robust Peer-to-Peer Networks: Search Methods", RFC 4981, September 2007.

[Ratnasamy] Ratnasamy, S., Francis, P., Handley, M., Karp, R., and S. Shenker, "A Scalable Content-Addressable Network", Proceedings of ACM SIGCOMM 2001, January 2001.

[Rowaihy] Rowaihy, H., Enck, W., McDaniel, P., and T. Porta, "Limiting Sybil attacks in structured peer-to-peer networks", Proceedings of IEEE INFOCOM 2007, May 2007.

[Rowstron] Rowstron, A. and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", Proceedings of 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), November 2001.

[SHA1] 180-1, FIPS., "Secure Hash Standard", April 2005.

[SKYPE] "Skype", <http://www.skype.com/>.

[Saxena] Saxena, N., Tsudik, G., and J. Yi, "Admission Control in Peer-to-Peer: Design and Performance Evaluation", Proceedings of 1st ACM workshop on Security of ad hoc and sensor networks, October 2003.

[Scheideler] Scheideler, C., "How to Spread Adversarial Nodes?: Rotate!", Proceedings of 37th Annual ACM Symposium on Theory of Computing, May 2005.

[Seedorf1] Seedorf, J., "Security Challenges for Peer-to-Peer SIP", IEEE Network, vol. 20, no. 5, September 2006.

[Seedorf2] Seedorf, J., "Using Cryptographically Generated SIP-URIs to Protect the Integrity of Content in P2P-SIP", Proceedings of 3rd Annual VoIP Security Workshop, June 2006.

[Singh] Singh, K. and H. Schulzrinne, "Peer-to-Peer Internet Telephony using SIP", Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video, June 2005.

[Sit] Sit, E. and R. Morris, "Security considerations for peer- to-peer distributed hash tables", Revised Papers from First International Workshop on Peer-to-Peer Systems, March 2002.

[Stoica] Stoica, I., Morris, R., Karger, D., Kaashoek, M., and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", Proceedings of Applications, Technologies, Architectures, and Protocols for Computer Communication 2001, May 2001.

[Tam] Tam, J., Simsa, J., Hyde, S., and L. Ahn, "Breaking Audio CAPTCHAs with Machine Learning Techniques", Proceedings of Advances in Neural Information Processing Systems, December 2009.

[Uzun] Uzun, E., Pariente, M., and A. Selpk, "A Reputation-Based Trust Management System for P2P Networks", Proceedings of International Symposium on Cluster Computing and the Grids, April 2004.

[Wallach] Wallach, D., "A Survey of Peer-to-Peer Security Issues", Proceedings of International Symposium of Software Security 2002, November 2002, <http://www.cs.rice.edu/~dwallach/pub/tokyo-p2p2002.pdf>.

[Yu] Yu, H., Kaminsky, M., Gibbons, P., and A. Flaxman, "SybilGuard: Defending Against Sybil Attacks via Social Networks", Proceedings of ACM SIGCOMM 2006, September 2006.

[Zhang] Zhang, X., Chen, S., and R. Sandhu, "Enhancing Data Authenticity and Integrity in P2P Systems", IEEE Internet Computing, vol. 9, no. 6, September 2005.

[Zimmermann] Zimmermann, Philip., "Pretty good privacy: public key encryption for the masses", Building in big brother: the cryptographic policy debate pag. 103-107, 1995.

Authors' Addresses

Henning Schulzrinne
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA

EMail: hgs@cs.columbia.edu

Enrico Marocco
Telecom Italia
Via G. Reiss Romoli, 274
Turin 10148
Italy

EMail: enrico.marocco@telecomitalia.it

Emil Ivov
SIP Communicator / University of Strasbourg
4 rue Blaise Pascal
Strasbourg Cedex F-67070
France

EMail: emcho@sip-communicator.org