# Security in P2P Networks: Survey and Research Directions

Esther Palomar, Juan M. Estevez-Tapiador,
Julio C. Hernandez-Castro, and Arturo Ribagorda

Computer Science Department – Carlos III University of Madrid
Avda. Universidad 30, 28911, Leganes, Madrid
{epalomar, jestevez, jcesar, arturo}@inf.uc3m.es

**Abstract.** A fundamental feature of Peer-to-Peer (P2P) networks is the honest collaboration among an heterogeneous community of participants. After `Napster` success –the first P2P file sharing application massively used–, advances in this area have been intense, with the proposal of many new architectures and applications for content and computing sharing, and collaborative working environments. However, the inherent differences between the P2P model and the classic client-server paradigm cause that many security solutions developed for the latter are not applicable or, in the best case, have to be carefully adapted. In this paper, we present a survey on security issues in P2P networks, providing a comparative analysis of existing solutions and identifying directions for future research.

## 1 Introduction

P2P is often described as a type of decentralized computing where nodes communicate directly with each other. P2P applications allow users to communicate synchronously, supporting tasks such as instant messaging, working on shared documents or sharing files, among many others. As a result, the P2P paradigm provides users with the capability of integrating their platforms within a distributed environment with a broad range of possibilities.

A P2P network has neither clients nor servers; each individual node could act simultaneously as a client and as a server for the rest of the nodes in the network. Within this paradigm, any node can initiate or complete a transaction, and it can also play an active role in the routing operations. In general, nodes will be users' personal computers, instead of typical elements of the network infrastructure, but they can present heterogeneous characteristics regarding the local configuration, processing power, connection bandwidth, storage capacity, etc.

Despite the advances in P2P technology, security-related issues have remained systematically unaddressed or, at best, handled without a global perspective [1]. Classic approaches have concentrated on specific points, such as providing anonymity to users and data [2], or on establishing and managing trust relationships among users. Research efforts have also focused on the study of Denial of Service (DoS) attacks and the abuse of multiple identities (*Sybil* attack) [3].

X. Zhou et al. (Eds.): EUC Workshops 2006, LNCS 4097, pp. 183–192, 2006.

Other problems have been recently identified, as those associated to the transience of peers (*churn*) or how to combat the selfish behavior exhibited by nodes that do not share their resources (*free-riding*).

The study of security issues in P2P networks becomes more difficult due to the diversity and heterogeneity of existing P2P architectures. With the aim of providing a general analysis, we have identified three elements common to every P2P system:

- The user community (nodes).
- The overlay architecture which defines the logical structure of the network over the underlying communication layer(s). Essentially, the overlay network manages the aspects related to node location and message routing.
- The information (content) stored at nodes and accessible through the services offered by the network.

The remainder of this article is organized as follows. In Section 2, we review the most significant papers and research studies for overlay networks and P2P secure routing, analyzing security approaches and their main limitations. Similarly, Sections 3 and 4 focus on user community management and content storage and distribution, respectively. Our research results are described in Section 5, where a comparative analysis of the security issues in the most relevant P2P architectures is shown. Finally, Section 6 discusses some open issues that could became interesting avenues for future research.

## 2 Overlay Networks

Basically, overlay networks are responsible for providing a resource location service. Overlay networks can be classified in terms of their degree of centralization and structure. There are three categories concerning the former:

- **Purely Decentralized Architectures:** There is no central coordination point of the distribution activities. Nodes are referred as *servents* due to their dual nature (SERVers+cliENTS).
- **Partially Centralized Architectures:** Special roles are assumed for some nodes called “supernodes”, which carry out special tasks mainly aimed at improving the performance of network routing.
- **Hybrid Decentralized Architectures:** A central server provides the interaction among the nodes, since indexes which support data searches and node identification are centralized, but the data is distributed.

On the other hand, P2P networks are categorized in terms of their structure as **unstructured** or **structured**. The first category of overlay models was popularized by `Napster`, which showed some scalability limits, but reduced the network dependence to a small number of highly connected, easy to attack peers. Peers join the network without any prior knowledge of the topology. Searching mechanisms include brute force methods, such as flooding the network with propagating queries to locate highly replicated contents. Other well-known unstructured systems are `Gnutella` and `FastTrack/KaZaA`. On the contrary, structured P2P networks provide a mapping between content and location in the form of a

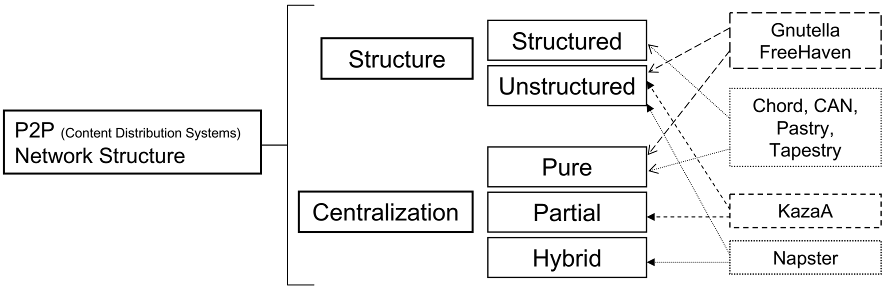

**Fig. 1.** Classification

distributed routing table. Queries can be efficiently routed to the node having the desired content, and data items can be discovered using the given keys. The overlay network assigns keys to data items and organizes its peers into a graph that maps each data key to a peer. Maintenance of this graph is not easy, especially due to the high transience of nodes. Figure 1 sketches both categories, and shows where some representative P2P systems fall.

## 3 User Community

In a P2P network, the user community is characterized by a high node transience, total ignorance of the node's intentions, and the lack of a centralized authority. These issues have been tackled by different models, protocols and systems. Next, we mention the most significant approaches.

### 3.1 Trust and Social Profiles

Cooperation plays an essential role in P2P networks, and evolution of that property is the creation of social profiles of P2P virtual communities [4], which are addressed by recent investigations focused on the establishment of incentives that motivate users to behave well. Typically, at the heart of these researches operate traditional reputation schemes as polling-based algorithms. A solution to encourage resource sharing is to force each peer to contribute before being served. This collaboration is evaluated for computing a user participation level, rewarding the most collaborative peers with, for example, high priorities for their queries or by decreasing the transmission delays of their desired services.

The model proposed in [5] manages utility functions for individual users as a function of the quality of service. These utility functions are mainly based on the amount of shared contents and their quality for estimating the node's aptitude. Unfortunately, the main drawback is that these kind of mechanisms are easily disrupted by the actions of dishonest nodes.

### 3.2 Identification vs. Anonymity

Node identification and its relationship with anonymity is an intense research area due to the potential risk of performing traffic analysis attacks and the traceability of communications among nodes. A correct node identification is critical, and the lack of control on it could yield to vulnerabilities in the replication process, ID spoofing, and DoS attacks (or DDOS, against which a solution such as that proposed in [6] can be applied). An example is the problem of *churn*, which involves a large number of potentially malicious peers in the P2P system to certify the peers identities. The simplest design to assign an ID to each node is to have a centralized authority providing cryptographic certificates, which is only consulted when new nodes join. Others approaches bind an e-mail address to a public key; in this way, some F2F systems' users –where a ring of trust is created– generate a public key (without CA) that is sent to the authenticated user by e-mail. This authentication does not exist in the majority of P2P systems, even though some studies show that it may be possible to use some form of cryptographic puzzles to avoid that attackers with a large amount of computational resources can get a huge range of node IDs.

Some P2P architectures use cryptographic techniques to prevent adversaries from observing or modifying network-level communications between legitimate nodes. We have to consider that the attacker is able to use these properties for linking messages and, correspondingly, the pseudonyms used with them. Onion routing provides application-independent, real-time, and bidirectional anonymous connections (not anonymous communications) that are resistant to both eavesdropping and traffic analysis. Some existing approaches on anonymity are provided by `Crowds` [7], `Hordes` [2], `Tarzan` [8], `Freedom`, and `FreeHaven` [9]. On the other hand, the idea of using pseudonyms rises solving attacks against anonymity, and also the use of (*blacklisting*). A digital pseudonym can be somehow linked to a public key with the aim of testing digital signatures. Pseudonyms attacks involves several scenarios such as *cheating* and *Sybil* attacks, and *free-riding*, where non-cooperative users benefit from others resources [10,11]. P2P systems will continue being vulnerable to these attacks due to the lack of public key infrastructures, but some recent approaches address them with techniques based on Byzantine agreements [12]. An alternative solution would require methods based on the use of *micropayments* mechanisms.

### 3.3 Node Authentication and Access Control

Concerning node authentication, a malicious node may take part in man-in-the-middle situations where it can send an unsolicited response to a query or can attempt to forge a message with incorrect results. The best defense against this would be to employ standard authentication techniques, such as digital signatures or message authentication codes. However, digital signatures are somewhat computationally expensive, and MACs require shared keys.

In the vast majority of systems proposed so far, the absence of authentication is solved by distributing appropriate keys into groups of authorized users for granting access to the shared content (see Section 4). `Oceanstore` is an example of a system in which each owner assigns contents an access control list (ACL) by using digital certificates. Every content alteration is verified against the ACL, ignoring non-authorized updates. A different approach is presented by Pathak and Iftode [12]. The lack of TTP (Trusted Third Party) motivate users to classify nodes into three categories: “trusted”, “untrusted” or “others”, after reaching an agreement through proofs of possession of a legitimate copy of the untrusted node’s public key. The other side of the spectrum is represented by applications where users are simply known, or “friends”, and share their friends with new friends, such as in *Friend-to-Friend* protocols (F2F) discussed in [13].

## 4 Content

Availability has a significant influence on popularity. In fact, it is probably the security property that user most worry about. Availability is measured by how often object requests are successfully served, and, in particular, mapping two factors: the number of peers (average node availability) and the number of object replicas (replica storage size).

### 4.1 Replication

The most commonly used file replication strategy in P2P systems simply makes replicas of objects on the requesting peer, upon a successful query/reply. Nevertheless, an important problem is how to deal with an overestimated number of copies that could cause serious security conflicts, like DoS attacks.

Some algorithms intends to increase the availability of all shared files toward a common level, while allowing peers to act completely autonomously by using only a small amount of loosely synchronized global data. In `Gnutella`, a decentralized P2P infrastructure has been implemented to hold self replication (false information distribution), man-in-the-middle, pseudospoofing, ID stealth and shilling attacks. It is necessary to balance the total network anonymity and the need of preventing network abuse, to assure content’s high quality and to earn server good reputation. In this system, it is considered the servent reputation (public key digest), the resource reputation (content digest), and a simple binary algorithm for voting. In the same way, `Freenet` and `Chord` do not assign responsibility for data to specific peers, and lookups take the form of searches for cached copies. However, in `Freenet` files are identified by content-hash keys, which gives every file a pseudo-unique data file key, and by secured signed-subspace keys, to ensure that only one owner can write to a file and anyone can read it.

Emergent security problems, such as *attrition* attacks, which perform a especial type of network flooding, are already addressed by some recent models, e.g. by LOCKSS (*Lots Of Copies Keep Staff Safe*) [14].

### 4.2 Content Integrity and Authentication

The integrity of information in a P2P system may be attacked through the introduction of degraded-quality content or by misrepresenting the identity of the content (e.g. falsely labeling).

So far, reputation systems try to avoid corruption attacks by enabling users to rate both the content validity and the content provider. To ensure that all copies of the same content share the same reputation, a content may be identified by a cryptographic hash. Furthermore, it is required to guarantee that an attacker cannot modify or delete its client's reputation information, so designers must distribute this information among other clients using protocols that prevent tampering. Since attackers can delete clients and reinstall new ones, a reputation system should also maintain information for the machines on which clients run. However, this does not fit well with anonymity. Finally, content authentication is commonly uncertain and current research efforts have adopted popularity-based ranking systems to help users discover desired contents.

A more generalized approach for preventing content alteration is to acquire several copies of a file from different sources using voting or selection schemes. An interesting approach is introduced by Dwork and Naor [10] to increase the cost of sending email and make sending spam unprofitable. This concept has been extended to more general settings, such as preventing network level DoS attacks for TCP [22]. We have mentioned the consequences when a node acts maliciously in flooding-based overlay topologies. Peers can stem the flood of requests by demanding each request be accompanied by requester proofs of work (e.g. solution of a cryptographic puzzle) [23]. An alternative to client puzzles is to use the reputation systems mentioned above to track individual machine's utilization of networks resources. Abadi et al. [24] contribute with an approach based on the application of memory-bound functions to discouraging spam, taking into account that these are much more platform-independent than CPU-bound functions.

## 5 Analysis

In this section, we analyze the security properties considered by the three categories identified above: overlay routing (Table 1), user community (Table 2), and content distribution (Table 3). Note that the study takes into account several

**Table 1.** Security properties considered by overlay architectures, according to if they apply detection methods(△), protection mechanisms (▽), or both (♢). Symbol (◯) indicates a deficient mechanism, while (–) stands for non-applicable.

| | **ARCHITECTURES AND SYSTEMS** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Structured** | | | | | | **Unstructured** | | |
| | [18] | [19] | [20] | [21] | [22] | [8] | KaZaA | [23] [9] | **GRID** [24] |
| **ID Assignment** | | | | | | | | | |
| ID spoofing | ♢ | △ | ♢ | ♢ | – | ♢ | – | – | – |
| Pseudospoofing | – | – | – | – | – | – | ◯ | ♢ | – |
| **Routing** | | | | | | | | | |
| Churn | ♢ | ▽ | – | ♢ | – | – | – | – | ♢ |
| DoS | ♢ | △ | ♢ | – | △ | △ | ♢ | ♢ | – |
| **Dishonest Nodes** | | | | | | | | | |
| Cheating | ◯ | ◯ | ◯ | ◯ | – | △ | – | ♢ | ♢ |
| Sybil | – | – | – | ♢ | – | △ | – | – | – |
| Man-in-the-Middle | – | ◯ | △ | – | – | △ | – | – | – |
| **Properties** | | | | | | | | | |
| Availability | ♢ | ◯ | ♢ | – | ♢ | – | ♢ | ♢ | ♢ |
| Integrity | – | ♢ | ♢ | ♢ | – | ♢ | – | ♢ | – |
| Authentication | – | ◯ | – | – | – | – | – | ♢ | ♢ |
| Confidentiality | – | – | ♢ | ♢ | – | ♢ | – | – | – |
| Anonymity | ◯ | ♢ | ◯ | ◯ | – | ♢ | ♢ | ◯/♢ | – |

**Table 2.** Security properties considered by P2P architectures and systems for user community management. (Same legend as in Table 1)

| | **Trust** | **Anonymity** | | **Authentication** | | |
|---|---|---|---|---|---|---|
| | [13], $\mu$-payments | [7], [2] | [8], [9] | Oceanstore, [12] | [13] | **Middleware** |
| **Traceability** | | | | | | |
| ID spoofing | ▽ | – | ◇ | – | – | ○ |
| Pseudospoofing | ▽ | ○ | ◇ | – | – | ○ |
| Man-in-the-Middle | ◇ | – | – | ◇ | ◇ | ○ |
| **Availability** | | | | | | |
| Attrition (DoS) | – | – | ◇ | △ | – | – |
| **Dishonest Nodes** | | | | | | |
| Cheating | – | ○ | ○ | – | – | ○ |
| Sybil | ○ | ○ | ○ | – | – | – |
| Free-riding | ○ | ○ | ○ | – | ○ | ◇ |
| **Fairness** | ◇ | – | – | ◇ | – | – |
| **Properties** | | | | | | |
| Availability | – | – | – | ○ | – | ◇ |
| Integrity | ◇ | ▽ | ▽ | ◇ | ◇ | – |
| Authentication | ◇ | – | – | ◇ | ◇ | ◇ |
| Confidentiality | ◇ | ◇ | ◇ | ◇ | ◇ | – |
| Anonymity | ○ | ◇ | ◇ | – | ○ | – |

**Table 3.** Security properties considered by P2P architectures and systems for content management. (Same legend as in Table 1)

| | **Storage and Replication** | | **Search and Retrieval** | | **Integrity and Authentication** | |
|---|---|---|---|---|---|---|
| | [5], [14] | [23]others | [23], [19], [18] | *Ranking*, Similarity | [16], [17] | Reputation S. |
| **Identification** | | | | | | |
| Pseudospoofing | ◇ | – | – | – | – | ○ |
| Blacklisting | – | ◇ | – | ◇ | – | ◇ |
| **Service Avail.** | | | | | | |
| DoS | – | – | ◇ | – | ▽ | – |
| Attrition | ◇ | – | ◇ | ○ | ▽ | – |
| Churn | ◇ | – | – | △ | △ | ○ |
| **Dishonest Nodes** | | | | | | |
| Cheating | ○ | ◇ | ◇ | ◇ | – | ◇ |
| Sybil | – | – | – | – | ▽ | – |
| Free-riding | ○ | – | ◇ | – | – | – |
| Man-in-the-Middle | – | ◇ | – | – | – | ◇ |
| Fairness | ◇ | – | – | ◇ | ▽ | – |
| **Properties** | | | | | | |
| Availability | ◇ | ◇ | ◇ | ◇ | – | – |
| Integrity | – | – | – | ◇ | – | ◇ |
| Authentication | – | △ | – | ◇ | ◇ | ◇ |
| Confidentiality | – | – | – | – | – | △ |
| Anonymity | ○ | ○ | ○ | ○ | – | ○ |

dimensions according to the structure, architecture and system affected by some security attacks. Thus, for each approach, we analyze the degree of detection and protection, even the absence, against the exposed attacks.

Each row in the tables corresponds to a particular class of attack, while columns indicate if a specific architecture, model or system implements mechanisms for defending against it. At first sight, from Table 1 it might seem that current works explore the benefits of enhanced request routing in P2P file sharing, most of them against DoS attack and ID spoofing. A significant proportion of the research efforts are essentially worried about availability and integrity properties. Current efforts in overlay are focused on authenticated query routing, while some of them are only studying the consequences of malicious actions and proposing protection models against cheating and Sybil attacks (e.g. `Gnutella` and `Tarzan`). Analogous difficulties arise in the real application of anonymity.

Table 2 contains the analysis of the peers' behavior at most popular reputation systems, anonymity architectures and application-specific models, showing that there is significant heterogeneity in peers traceability, availability, and vulnerabilities. Confidentiality, anonymity and integrity are not taken into account in most systems. Nevertheless, to understand which issues account for these unavailability misses, we first explored the relationship between the protection against DoS attacks provided by main P2P overlays, obtaining poor matches. Unfortunately, in any case the protection against dishonest nodes manipulations are devoid of any detection mechanism. Fairness begins to be taken into account to control aggressive behavior ("antisocial") between peer connections.

Concerning content management, since the performance is sensitive to the degree of user cooperation, it makes sense to provide incentives to users to share their resources. In particular, an adequate option would be to increase their download allocations in a manner that depends on their contributions. As a result, most P2P systems manage efficiently the content availability and a fair sharing (see Table 3). However, anonymity is less considered and, therefore, many attacks are not applicable. This uncertainty is not worrisome, for Table 3 does not include systems based on anonymity protocols. We have thoroughly examined the range of activities performed by dishonest nodes against almost all the security properties of the content. Based on these results, we can conclude that every system discussed in Table 3 is vulnerable, at least, to one of the mentioned attacks.

## 6 Conclusions and Research Directions

Current P2P networks present a number of security problems in adaptability, self-management, scalability, fault-resilience in the presence of network and computing failures, and availability in the presence of peers' transience. Most of these concerns are applicable at several levels, such as those studied in this paper. Future work would focus on extensions to the following items:

- Topics related to node cooperation and fairness among virtual societies (communities) and social profiles, node integrity protocols, and recently content authentication protocols.
- P2P systems for mobile and ad-hoc networks introduce a number of new issues related to naming, discovery, communication and security. In particular, these systems require lightweight and efficient architectures due to the highly dynamic and constrained nature of these environments.

- The idea of using cryptographic puzzles for decreasing spam is being extended to P2P networks. This idea could provide access control and detect DoS attacks in advance.

P2P networks are useful not only for relatively simple file sharing systems, in which the main goal is directly exchanging contents with others. However, large P2P distribution networks will be more robust against attacks and range to more sophisticated structures which self-organize into network topologies with the purpose of sharing resources such as content and CPU cycles, of maintaining secure and efficient storage, indexing, searching, updating, and retrieving data. We performed the present study of P2P content distribution systems and infrastructures by identifying the feature space of their functional and non-functional characteristics linking them to current security challenges (anonymity, fairness, scalability, performance, content management, etc.) and without forgetting emergent applications such as MANET, GRID, and collaborative environments.

We have presented a survey of existing security approaches in P2P networks according to the P2P architecture adopted. It was proposed to categorize the most popular protocols depending on how they detect and protect against various attack scenarios. Our analysis summarizes security characteristics adopted by those P2P structures.

## References

1. Balfe, S., Lakhani, A., Paterson, K.: Trusted computing: Providing security for peer-to-peer networks. In: Proc. 5th IEEE Int. Conf. Peer-to-Peer Computing, Konstanz, Germany, IEEE Press (2005) 117–124
2. Levine, B., Shields, C.: Hordes: A protocol for anonymous communication over the internet. Computer Security **10** (2002) 213–240
3. Douceur, J.: The sybil attack. In: Proc. 1st Int. Workshop on Peer-to-Peer Systems, Cambridge, USA (2002) 251–260
4. Sakaryan, G., Unger, H., Lechner, U.: About the value of virtual communities in p2p networks. In: Proc. 3rd Int. School and Symposium, Mexico (2004) 170–185
5. Cuenca-Acuna, F., Peery, C., Martin, R., Nguyen, T.: Planetp: Using gossiping to build content addressable peer-to-peer information sharing communities. In: Proc. 12th IEEE Int. Symp. High Performance Distributed Computing, Washington,USA, IEEE Press (2003) 236–246
6. Lee, F.Y., Shieh, S.: Defending against spoofed ddos attacks with path fingerprint. Computers & Security **24** (2005) 571–586
7. Reiter, M., Rubin, A.: Crowds: Anonymity for web transactions. ACM Transactions on Information and System Security **1** (1998) 66–92
8. Freedman, M., Morris, R.: Tarzan: a peer-to-peer anonymizing network layer. In: Proc. 9th ACM Conf. Comp. and Comm. Sec., Washington (2002) 193–206
9. Dingledine, R., Mathewson, N., Syverson, P.: The free haven project: Reputation in p2p anonymity systems. In: Proc. Int. Workshop Design Issues in Anonymity and Unobservability, Berkeley, USA (2003)
10. Dwork, C., Naor, M.: Pricing via processing or combatting junk mail. In: Proc. CRYPTO'92. Volume 740 of LNCS., Springer-Verlag (1992) 139–147

11. Feldman, M., Chuang, J.: Overcoming free-riding behavior in peer-to-peer systems. ACM Sigecom Exchanges **6** (2005) 41–50
12. Pathak, V., Iftode, L.: Byzantine fault tolerant public key authentication in peer-to-peer systems. Computer Networks **50** (2006) 579–596
13. Edwards, W.: Using speakeasy for ad hoc peer-to-peer collaboration. In: Proc. ACM Conf. Computer Supported Cooperative Work, New Orleans (2002) 256–265
14. Maniatis, P., Giuli, T., Roussopoulos, M., Rosenthal, D., Baker, M.: Impeding attrition attacks in p2p systems. In: Proc. 11th ACM SIGOPS European Workshop, Leuven, Belgium, ACM (2004)
15. Ratnasamy, S.: A scalable content-addressable network. Technical report, Berkeley (2002)
16. Stoica, I., Morris, R., Karger, D., Kaashoek, M., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: Proc. ACM SIGCOMM Conference, San Diego, USA (2001) 149–160
17. Rowstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Proc. IFIP/ACM Int. Conf. Distributed Systems Platforms (Middleware), Heidelberg, Germany (2001) 329–350
18. Zhao, B., Kubiatowicz, J., Joseph, A.: Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical report (2001)
19. Kim, Y., Lau, W., Chuah, M., Chao, J.: Packetscore: Statistical-based overload control against distributed denial-of-service attacks. In: INFOCOMM'04, Hong Kong, China, IEEE Press (2004)
20. Defigueiredo, D., Garcia, A., Kramer, B.: Analysis of peer-to-peer network security using gnutella. Technical report (2002)
21. Anderson, D., Cobb, J., Korpela, E., Lebofsky, M., Werthimer, D.: Seti@home: An experiment in public-resource computing. Comms. of the ACM **45** (2002) 56–61
22. Zhou, F., Zhuang, L., Zhao, B., Huang, L., Joseph, A., J.Kubiatowicz: Approximate object location and spam filtering on peer-to-peer systems. In: Proc. ACM Int. Middleware Conf., Rio de Janeiro, Brazil, ACM (2003) 1–20
23. Juels, A., Brainard, J.: Client puzzles: A cryptographic defense against connection depletion attacks. In: Proc. NDSS'99, California (1999) 151–165
24. Abadi, M., Burrows, M., Manasse, M., Wobber, T.: Moderately hard, memory-bound functions. ACM Transactions on Internet Technology **5** (2005) 299–327