Leveraging Social Links for Trust and Privacy in Networks

Leucio Antonio Cutillo*, Refik Molva*, and Thorsten Strufe ‡

* EURECOM, Sophia-Antipolis, France
‡ TU Darmstadt, Darmstadt, Germany
{cutillo,molva,strufe}@eurecom.fr

Abstract. Existing on-line social networks (OSN) such as Facebook suffer from several weaknesses regarding privacy and security due to their inherent handling of personal data. As pointed out in [4], a preliminary analysis of existing OSNs shows that they are subject to a number of vulnerabilities, ranging from cloning legitimate users to sybil attacks through privacy violations. Starting from these OSN vulnerabilities as the first step of a broader research activity, we came up with a new approach that is very promising in re-visiting security and privacy problems in distributed systems and networks. We suggest a solution that both aims at avoiding any centralized control and leverages on the real life trust between users, that is part of the social network application itself. An anonymization technique based on multi-hop routing among trusted nodes guarantees privacy in data access and, generally speaking, in all the OSN operations.

1 Introduction

According to the authors together with a set of studies like [4], [12], [2], [9], [13], OSN applications seem to suffer from data and communications privacy exposures that call for 'classical' mechanisms such as data confidentiality, integrity, authentication, access control, data hiding and data minimization. Exploiting such exposures is not impossible at all, as demonstrated in [4], where authors manage to access a large volume of user data by creating and launching automated crawlers against popular social networking sites. To make things worse, a careful analysis of the privacy problem with current OSNs shows that even if a complete set of security and privacy measures were deployed, current OSN would still be exposed to potential privacy violations by the omniscient Social Network Service (SNS) provider, that becomes a big brother, as every user's data (messages, profiles, relations) is collected and stored in a centralized way. Current OSN services are not likely to address this problem in a near future

This work has been supported by the SOCIALNETS project, grant agreement number 217141, funded by the EC seventh framework programme theme FP7-ICT-2007-8.2 for Pervasive Adaptation.

since access to users' private data is the underpinning of a promising business model, as one can see e.g. from the virtual value of these SNS providers, that in the case of Facebook arises to 15 billion US$ according to its deal with Microsoft in 2007 (see [1]).

As the first objective of our approach we thus claim that user privacy in OSN systems can only be assured through the avoidance of centralized control by an omniscient authority. At this purpose, infrastructure-less peer-to-peer model seems to be a natural base to build a solution that avoids this centralized control, although a corollary of the peer-to-peer model on the other hand is the lack of a priori trust among parties, which comes as an additional requirement. As the second strong point of our approach we suggest that trust in communications and distributed computing can be built based on the trust relationships that are akin to the social network. Thus, network nodes operated by people who are friends in the social network can leverage on this friendship relation to build trusted channels or to enforce cooperation in a self-organizing system such as an ad hoc network or a peer-to-peer application.

The rest of this paper is divided into six sections: in section 2 we suggest Safebook, a privacy-preserving distributed architecture for an on-line social network that has been sketched in [7]. The design of Safebook is governed by two principles:

- avoiding centralized control through a de-centralized peer-to-peer architecture;
- leverage social trust relationships from the social network in achieving security and privacy as part of the social network system.

Safebook mostly focuses on data storage and lookup functions and provides each user private data storage in trusted nodes based on the application-specific trust relationships. In order to prevent intruders or the network provider(s) from violating users' privacy, some anonymous communication techniques leveraging the social trust relationships are also integrated: each hop corresponds, in fact, to a real life friendship link, thus both enhancing hop-by-hop cooperation and reducing the presence of malicious nodes in the communication paths.
The feasibility aspects of Safebook are investigated in section 3, that also presents some preliminary analysis of Safebook's performances according to time and data availability questions.
Section 4 presents the related work covering the research domain of peer-to-peer OSNs.
Finally, section 5 presents the conclusions of this work.

2 Architecture

The architecture of Safebook consists of two overlays, as shown in fig.1. Each Safebook node is thus part of the Internet, the peer-to-peer overlay and the social network overlay. The components of Safebook (cmp. fig.1) are:

1. several ***matryoshkas***

2. a ***peer to peer substrate*** (e.g. a DHT)
3. a ***trusted identification service***

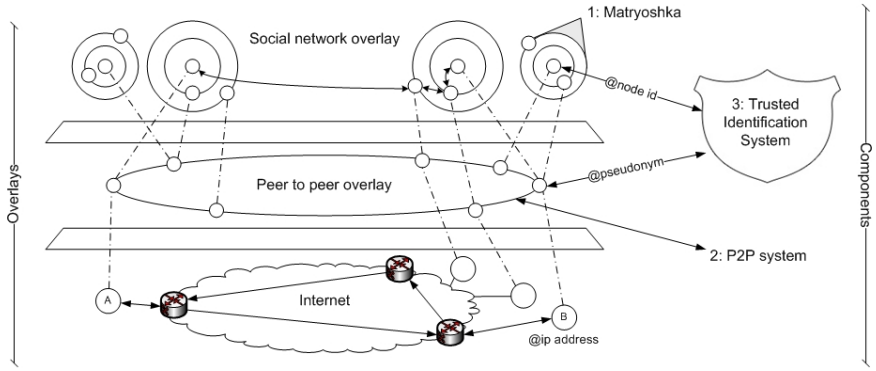

Fig. 1. Overlays of Safebook.

Matryoshkas are particular structures providing end-to-end confidentiality and distributed data storage with privacy. They leverage on existing trust of OSN members in real life. The Peer-to-peer substrate provides a decentralized global data access. The trusted identification service guarantees authentication and provides unique addresses to each member of Safebook. It can be provided off-line and may be implemented in a distributed fashion.

Matryoshkas The Matryoshka of a user is a structure composed by various nodes surrounding the user's node in concentric shells. The user's node is thus the ***core*** of his matryoshka and can also be part of some other users' matryoshkas. The inner shell of a matryoshka consists of nodes belonging to the trusted contacts of the user. The second shell consists of nodes that are trusted contacts of nodes in the inner shell and so on. It is important to note that nodes on the same shell do not necessarily share trust relationships between themselves, except for the inner shell, which all share their relation to the core node.

The nodes on the inner shell cache the data for the core and serve requests if the core is offline. A data request message reaches a node in the inner shell from a node in the outer shell through a path that provides hop-by-hop trust. The reply follows the same path in the reverse direction. Based on this, the matryoshkas assure cooperation enforcement in our OSN. We point out that the trust relationship between nodes is not used in a transitive fashion, as none of the nodes on a path, other than the direct neighbors, needs to be trusted by any user.

Peer-to-peer substrate The peer-to-peer substrate consists of all the nodes and provides data lookup services. Currently, a DHT based on KAD[11] is used as the P2P substrate. Nodes are arranged according to their pseudonyms and

lookup keys correspond both to members' node identifiers and to the hash of their attributes, like full names or the likes. All nodes that belong to the outer shell of a user's matryoshka register themselves as entrypoints for this matryoshka with the nodes that are responsible for the respective lookup keys. The identity of a peer is revealed only to his trusted contacts since they are the only ones that can link his IP address to his node identifier.

Trusted identification service The trusted identification service (TIS) guarantees resistance against sybil and impersonation attacks by providing each node with a unique pseudonym and node identifier, and the related certificates. The existence of the TIS does not contrast our goal of privacy preservation through decentralization since the TIS is not involved in any data management activity and it is used only to prevent impersonation and a free selection of a pseudonym and hence their position in the DHT. Moreover the TIS can be implemented in a decentralized fashion and does not have to be constantly online.

2.1 Operations

The most important operations of our OSN are the matryoshka creation, the profile publication and the data retrieval.

Matryoshka creation In order to join Safebook a member $\mathcal{V}$ has to be invited by another member $\mathcal{U}$. After this phase, having obtained the necessary credentials from the TIS, $\mathcal{V}$ can start building his matryoshka. $\mathcal{V}$'s final goal is to register in the DHT his node id and a particular set of lookup keys associated to his identity, as e.g. a hash of his full name[1]. At the beginning $\mathcal{V}$ has only $\mathcal{U}$ in his contact list, so he sends $\mathcal{U}$ a signed registration request containing the lookup key(s) he wants to register, his certificate associated to his node id signed by the TIS, and a time-to-live (ttl) counter. This first message presents the node id of the sender instead of his pseudonym. This prevents the node in the DHT responsible for $\mathcal{V}$'s lookup key from linking that key with $\mathcal{V}$'s pseudonym.

Once $\mathcal{U}$ receives the registration message it decreases the ttl counter, chooses one (or several) of his trusted contacts, called $\mathcal{W}$, as a next step and sends $\mathcal{W}$ the request message signed with his pseudonym. This will prevent the registering node in the DHT from retrieving the social relationships between the OSN members constituing $\mathcal{V}$'s matryoshka. It is important to note that no assumption is held about social relationship between $\mathcal{V}$ and $\mathcal{W}$. This process runs until the ttl counter expires, when $\mathcal{V}$'s lookup key is registered in the DHT. The node responsible for that key maintains a reference table associating the key with the ip addresses of the nodes belonging to the outer shell of $\mathcal{V}$.

The number of contacts each node chooses to forward the registration request is determined by the ***spanning factor***. It defines the branching of the tree through the matryoshka whose root is the core and whose leaves are the nodes in the outer shell, starting from the core's direct connections. The higher the

[1] $\mathcal{V}$ can of course choose to register different lookup keys, in addition to his node id, to increase his visibility.

spanning factor, the higher is the number of nodes composing the tree, and the higher is thus the probability to have a valid path through the tree, i.e. a path where all the nodes are online. The spanning factor and the number of inner shell nodes each core should have is fundamental to guarantee data availability and will be investigated in section 3.

Profile publication A user's data can be public, protected or private. Private data is only stored by the owner, while public and protected data are stored by the contacts being in the inner shell of the user's matryoshka. All the published data is signed by the owner and encrypted using a simple group-based encryption scheme.

Each node can manage the profile information, the trusted contact relations and the messages. The profile information consists of the data a member wants to publish in the OSN and is organized in atomic attributes. The trusted contact relations represent the *friend list* of the user and associate each contact with a particular trust level. The messages can be exchanged by each member of the OSN, in this case the communication doesn't stop at the first matryoshka shell but reaches the core.

Data retrieval The requests are routed according to the P2P protocol until they reach the node responsible for the lookup key. It sends back the list of all the nodes constituing the outer shell of the target node's matryoshka. The requesting node then sends its request to a subset of the outer shell nodes of the target matryoshka. The requests are forwarded through the matryoshka to the inner shell, whose nodes serve it and send a response along the inverse path.

3 Feasibility

In this section we will analyze the feasibility of our approach with respect to data availability and delays.

We will focus on:

- the minimum number of contacts a node needs to have in order to guarantee the availability of his data;
- the minimum number of hops in the matryoshkas to provide anonymity;
- the expected delay for data retrieval.

Data availability We can see each core as a root of a tree whose leaves lie in the outer shell. Let *nop* be the probability of each node being online, *span* the spanning factor of the tree passing through a user $\mathcal{V}$'s matryoshka and *shell* its shells number, i.e. the number of hops between $\mathcal{V}$ and whichever node in the outer shell. Let Λ be the set of all the inner shell nodes and $\|\Lambda\|$ its cardinality. Thanks to a simple geometric law (1) it is possible to compute the probability ov_{shell} that at least one inner shell node can be reached, i.e. the probability that

$\mathcal{V}$'s data is accessible.

$$\begin{aligned} ov_0 &= nop \\ ov_j &= nop\left(1-(1-ov_{j-1})^{span}\right), j \in [1 \dots shell-1] \\ ov_{shell} &= \left(1-(1-ov_{shell-1})^{\|\Lambda\|}\right) \end{aligned} \tag{1}$$

Let the probability to have at least one valid path through a user's matryoshka be as high as 90% as a requirement. We refer to a *valid path* as a path where each node is on-line. Assuming that $span = 1$, this goal is achieved with different values of *shell*, *nop*, and number of contacts in the inner shell, as shown in figure 2.

According to a recent work on Skype[2][8] we can assume *nop* to be at least as high as 0.3. We rely on this data since Skype, as Safebook, enhances users' interactions by providing messaging services such as chat.

As one can see in figure 2, the number of contacts in the inner shell λ that is needed with $shell = 3$ and $nop = 0.3$ is 85. With $shell = 4$ the number of these contacts increases to 290. By selecting a spanning factor of $span = 2$, the same availability is achieved with 13 to 23 contacts, respectively with $shell = 3$ and $shell = 4$ (see figure 3). This amount of contacts is much more likely to be reached. From previous studies we have access to the graph of Xing[3] and could show that the average number η of a member's contacts in that application is 24.

Minimum number of hops in matryoshkas Let's suppose a member $\mathcal{A}$ has a matryoshka with a single shell ($shell = 1$). Let's also suppose that a requester $\mathcal{B}$ knows this fact. $\mathcal{B}$ can perform a lookup on the P2P substrate and get the list Ω of the pseudonyms of all the nodes located on the outer shell of $\mathcal{A}$'s matryoshka, together with their IP addresses. In this case these pseudonyms belong to a subset of $\mathcal{A}$'s friends and $\mathcal{B}$, that can have by chance some of them in his own friend list, could find their identity.

Now let's suppose that $shell = 2$ and that $\mathcal{B}$ knows about it. If $\mathcal{B}$ had, by chance, some $\omega_j \in \Omega$ in his friend list $\mathcal{B}$ would have access to ω_j's friend list and be able to determine which one of ω_j's friends is a direct contact of $\mathcal{A}$. The probability for $\mathcal{B}$ to know all $\omega_j \in \Omega$ and their contacts in order to retrieve all the contacts λ_j in the inner shell Λ of $\mathcal{A}$'s matryoshka is

$$p_\Lambda = \left(\frac{1}{\eta}\right)^{\|\Omega\|}$$

where $\|\Omega\|$ is the cardinality of Ω and $span = 1$. This probability is negligible, but the probability of finding one contact, that is

$$p_{\Lambda,1} = 1 - \left(1 - \frac{1}{\eta}\right)^{\|\Omega\|}$$

[2] http://www.skype.com
[3] http://xing.com

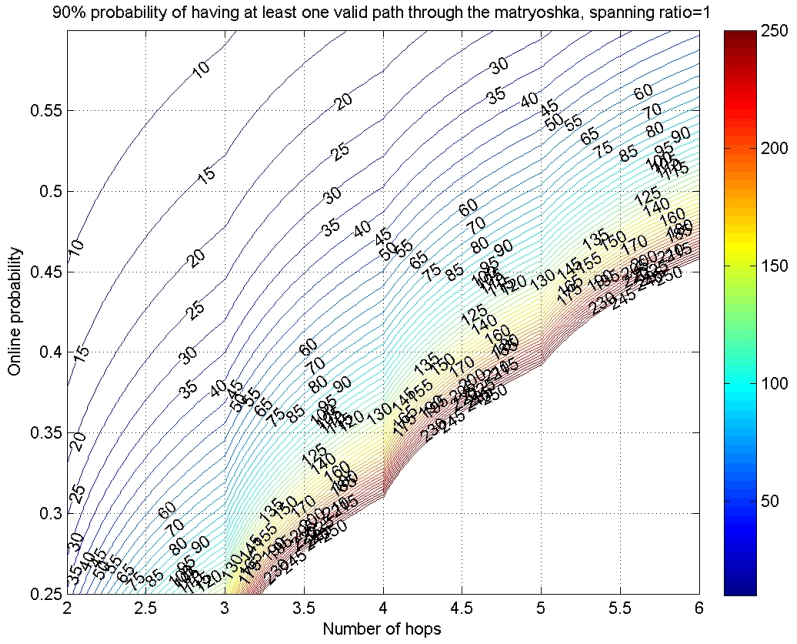

Fig. 2. Access data of a user - span=1.

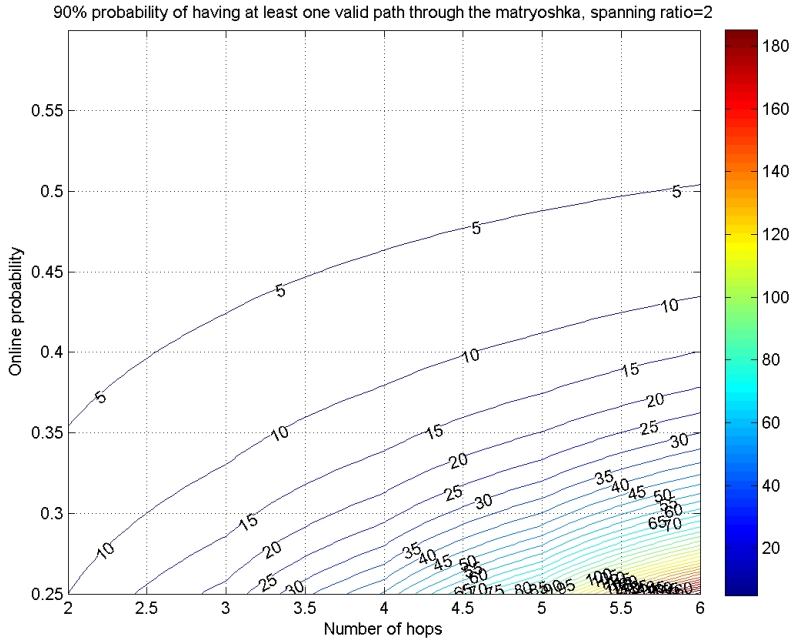

Fig. 3. Access data of a user - span=2.

is on the other hand quite large. However by increasing the number of shells both probabilities drastically decrease. Furthermore, as discussed above, in a realistic operational setting *span* has to be at least 2. Thus with $span \geq 2$, $\|\Omega\|$ increases exponentially with the number of shells due to the fact that $\|\Omega\| = span^{shell-1}\|\Lambda\|$, and both p_Λ and $p_{\Lambda,1}$ would decrease even faster than in the previous scenario. A number of 3 to 4 shells is thus not only feasible to assure data availability, but also a reasonable choice to provide anonimity.

Data lookup The overall data lookup time T_{dr} can be seen as the sum of the DHT lookup time T_{DHT} and the round trip time in the matryoshka T_M: the first one depends above all on the DHT, while the second one depends above all on the availability of nodes constituing the matryoshka itself.

The choice of the P2P substrate plays an essential role in our OSN performances since it determines T_{DHT}. Of all exiting DHTs we use Kademlia [11] due to its short response time. According to recent studies [16] conducted on KAD as implemented in aMule, 90% of the lookups succeed in less than four hops, while the median lookup latency is 5.8 seconds. The authors show that with a simple tuning of KAD parameters it is easy to decrease this value to 2.3 seconds. Moreover the median lookup time can be further on decreased by slight protocol modifications.

The round trip time in the matryoshka T_M can be seen as twice the time required to reach an inner shell node from an outer shell one. As we have shown in the previous sections, a number of hops between three and four reasonably guarantees to each member both anonimity and data availability. This number of hops is comparable with that one encountered, on average, for successful lookup in KAD. Being all the nodes in the matryoshkas also in the P2P substrate we can therefore assume $T_M \sim T_{DHT} = 2.3$.

Overall data lookup time T_{dr} is thus likely to be on the order of 5 seconds, without taking into account that the social proximity can correspond to the geographical one.

4 Related work

A number of work has been done to guarantee security and privacy in P2P networks. A survey of anonymous P2P networks is presented in [15] and [6]. anonbib[4] additionally gives a good overview of existing approaches.

Like Safebook, Freenet [10] protects the anonymity of data authors and readers thanks to hop by hop routing. In Safebook, however, each hop corresponds to a real life friendship link. This enhances hop by hop cooperation and thus reduces the presence of malicious nodes in communication paths. In Freenet nodes join the system by connecting to one or more existing nodes whose addresses are obtained out of band. An almost similar approach is present in Safebook, where the very first time a node joins the system it needs an invitation from another existing member. While Freenet can be seen as a cooperative distributed

[4] http://freehaven.net/anonbib

filesystem, where nodes lying on a path cache the data provided as an answer to the requester, in Safebook members' data is cached only by a selected subset of friends, thus decreasing the overall number of replica without penalizing data access, as explained in section 3.

Similarly to Freenet, GNUnet [3] aims at anonymous P2P networking thanks to indirection techniques. However GNUnet adopts flooding, that introduces intolerable delays for an online social network application like Safebook.

The performances of several P2P systems can be improved by creating groups of interest, where information about particular resources is more detailed and reliable. However, these groups of interest do not represent real life social groups, whose links are used by Safebook in order to build the matryoshkas.

PROSA [5] and Bittella [14] are examples of this approach. They improve the data retrieval by addressing data requests to peers sharing the same interests. In PROSA both the shared data and the queries are represented as vectors and their distance is used to selectively forward queries or provide data. In Bittella the peers' affinity is computed according to past file transfers and query matches. Safebook does not use this semantic-based approach since, as an OSN, lookup data represents the profile data of members rather than documents, as it happens in file sharing applications. Moreover, unlike [5] and [14], Safebook can not be built on top of a P2P network with flooding due to the too strict responsiveness requirement of an online social network application.

5 Conclusions

In this paper we presented the architecture and some preliminary evaluation of Safebook that is a decentralized social network designed with the main goal of preserving users' privacy with respect to potential intruders and avoiding centralized control by omniscient service providers. One of the underpinnings of this architecture is the fact that it extensively capitalizes on the characteristics of social networks in real life that it is aiming at supporting through its services. Thus, thanks to trust relationships that are inherent to the social network, Safebook is able to build trusted connections among nodes that assure data and communication privacy. Furthermore the focus of Safebook is privacy of users within the social network application leaving aside some generic communication privacy requirements such as anonymous communication in the face of a global network monitor. Nonetheless, the basic approach taken by Safebook, namely leveraging social characteristics such as trust in addressing data and and communication privacy, can be applied in a number of areas of network security, including anonymous communications. Peer-to-peer anonymous communication systems based on mixes and onion routing severely suffer from high cost and complexity that become prohibitive with the lack of incentives akin to the self-organized context. Revisiting anonymous communication techniques such as mixes and onion routing under the light of social links viewed as a new feature to create incentives and assure hop-by-hop privacy seems to be a promising approach. Similarly, another interesting enhancement with the same principle is security in ad hoc networks

whereby social links can provide a good base for solving the lack of a priori trust akin to self-organized environments.

References

1. Modelling The Real Market Value Of Social Networks. http://www.techcrunch.com/2008/06/23/modeling-the-real-market-value-of-social-networks/, 2008.
2. Sophos Facebook ID Probe. http://www.sophos.com/pressoffice/news/articles/2007/08/facebook.html, 2008.
3. K. Bennett and C. Grotho. Gap - practical anonymous networking. In *Privacy Enhancing Technologies workshop. Springer-Verlag, LNCS 2760*, pages 141–160, 2003.
4. L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks. 2008. WWW 2009, Madrid.
5. V. Carchiolo, M. Malgeri, G. Mangioni, and V. Nicosia. Prosa: P2p resource organisation by social acquaintances. pages 135–142.
6. T. Chothia and K. Chatzikokolakis. A survey of anonymous peer-to-peer file-sharing. In *Network-Centric Ubiquitous Systems*, pages 744–755. Springer.
7. L. A. Cutillo, R. Molva, and T. Strufe. Privacy preserving social networking through decentralization. In *Wireless On-demand Network Systems and Services*, Feb 2009.
8. S. Guha, N. Daswani, and R. Jain. An experimental study of the skype peer-to-peer voip system. In *Peer-to-Peer Systems*. Microsoft Research.
9. G. Hogben. Security issues and recommendations for online social networks. Technical Report 1, 2007.
10. B. W. Ian Clarke, O. Sandberg and T. W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Design Issues in Anonymity and Unobservability*, pages 46 – 66, 2000.
11. P. Maymounkov and D. Mazieres. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *P2P-Systems*, volume 2429, pages 53 – 65, 2002.
12. S. Moyer and N. Hamiel. Satan is on My Friends List: Attacking Social Networks. http://www.blackhat.com/html/bh-usa-08/bh-usa-08-archive.html, 2008.
13. A. Poller. Privatsphärenschutz in Soziale-Netzwerke-Plattformen. Fraunhofer SIT Survey, www.sit.fraunhofer.de, 2008.
14. I. M.-Y. R. Cuevas, C. Guerrero and C. Navarro. Bittella: A novel content distribution overlay based on bittorrent and social groups. In *Peer to Peer Networks*, Nov. 2007.
15. M. Rogers and S. Bhatti. How to disappear completely: a survey of private peer-to-peer networks. In *Sustaining Privacy in Autonomous Collaborative Environments*, 2007.
16. M. Steiner, D. Carra, and E. W. Biersack. Faster content access in KAD. In *Peer-to-Peer Computing*, Sep 2008.