# Safebook: Feasibility of Transitive Cooperation for Privacy on a Decentralized Social Network

Leucio Antonio Cutillo Refik Molva Thorsten Strufe
Eurécom
Sophia Antipolis
France
Email: {cutillo, molva, strufe}@eurecom.fr

## Abstract

*Social networking services (SNS), which provide the application with the most probably highest growth rates in the Internet today, raise serious security concerns, especially with respect to the privacy of their users. Multiple studies have shown the vulnerability of these services to breaches of privacy and to impersonation attacks mounted by third parties, however the centralized storage at the providers of SNS represents an additional quite significant weakness that so far has not satisfyingly been addressed.*

*In this paper we show the feasibility of "Safebook", our proposal for the provision of a competitive social networking service, which solves these vulnerabilities by its decentralized design, leveraging on the real life relationships of its users and means of cryptography.*

## 1 Introduction

Online social networks (OSN) represent real life relations in computer networks. All the data constituing both the network structure and its contents is stored by Social Network Services (SNS), which are run by commercial providers like the LinkedIn corp., xing AG, facebook, google, MySpace Inc. and the likes. Registered users can access these networks and use them in order to share private information, like pictures, videos and contact details, or professional ones, like CVs and work experience. Even if at a first glance OSNs offer a powerful solution for users to advertise themselves, serious concerns about their security and privacy arise due to their centralized architecture and their approach to the acceptance of their user's rights [1]. Users' data is not only insufficiently protected against malicious attacks [8, 11, 2] but access to it may be sold both inside the OSNs themselves, e.g. in the form of premium account offers, or aggregated to the social network as a whole, as it happened e.g. with myspace, bought by news corp. for 580 million US$ in 2005.

We propose Safebook, which is a new approach to online social networking that is based on two design principles:

- a peer-to-peer architecture in order to avoid control over user data and behaviour by a single entity such as the service provider;
- privacy and trust management for user data and communications in the OSN system leveraging the trust relations from the social network.

Safebook, as has been shown in [5], by design meets a broad set of security requirements whose relevance is also gathered from a series of studies [8, 11]: end-to-end confidentiality, authentication, access control, privacy, data integrity and data availability.

**End-to-end confidentiality** has to guarantee that no other than requesting and responding parties can access the exchanged data, so that eavesdropping is impossible. Since in Peer to Peer (P2P) systems messages are forwarded along a path of peers that possibly contains a malicious entity, a special focus has to be put on **man in the middle** attacks, as they may be easy to mount in this environment.

Proper **authentication** of members is required in order to achieve **access control**. Fine grained access control based on profile attributes to private data can be used to guarantee data disclosure according to trustworthiness of the requester.

**Privacy** aims at anonymity, unlinkability and untraceability of user communications as well as the confidentiality of personal information with respect to intruders and the system provider.

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.

[1] http://consumerist.com/5150175/facebooks-new-terms-of-service-we-can-do-anything-we-want-with-your-content-forever

978-1-4244-4439-7/09/$25.00 ©2009 IEEE

**Data integrity** aims at preventing tampering with profile data and **data availability** property represents one of main usability requirements. It guarantees that profiles can be accessed at any time and messages can be delivered to any user at any time, preventing **denial of service** attacks.

Even though Safebook meets the aforementioned requirements, the feasibility of the decentralized approach in terms of availability of data and responsiveness of the system remains an open question. In this paper we thus extend our work and analyze the performance of Safebook in relation to these two key aspects. Starting from detailed studies on well known social network platforms and considering the architecture of our new social network service we present a feasibility study on data availiability and response times of Safebook.

This paper is divided into five sections: section 2 is an overview of Safebook and section 3 presents a detailed feasibility study taking into account key factors like obfuscation and data availability vs delay and resource requirements. Section 4 presents the main related work covering the research domain of P2P Online Social Networks. Section 5 presents conclusions.

## 2 Cooperative Social Networking Service

Safebook is a decentralized privacy preserving on-line social network. Its design is governed by the objective of avoiding centralized control over user data and behaviour by service providers. As its second design principle, Safebook leverages trust relationships from the social network application in building secure communication and data management services in order to meet the requirements presented in section 1.

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.

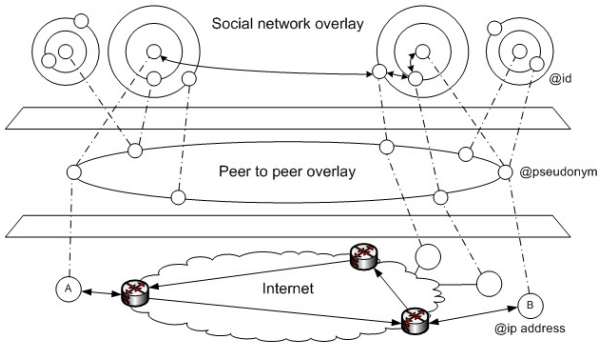


**Figure 1. Overlays of Safebook.**

The components of Safebook (cmp. fig.2) are:

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

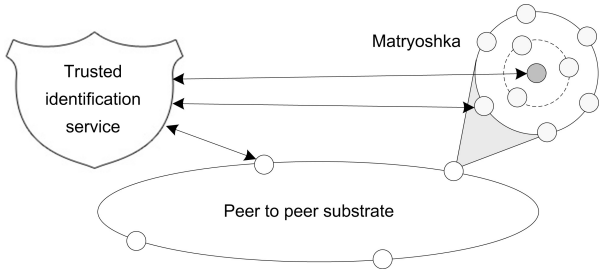


**Figure 2. Components of Safebook.**

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[10] 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[2]. 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 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 study

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

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

whichever node in the outer shell. Let $\Lambda$ 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 online. 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 3.

According to a recent work on Skype[3][6] 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 3, the number of contacts in the inner shell $\lambda$ 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 4). This amount of contacts is much more likely to be reached. From previous studies we have access to the graph of Xing[4] and could show that the average number $\eta$ of a member's contacts in that application is 24.

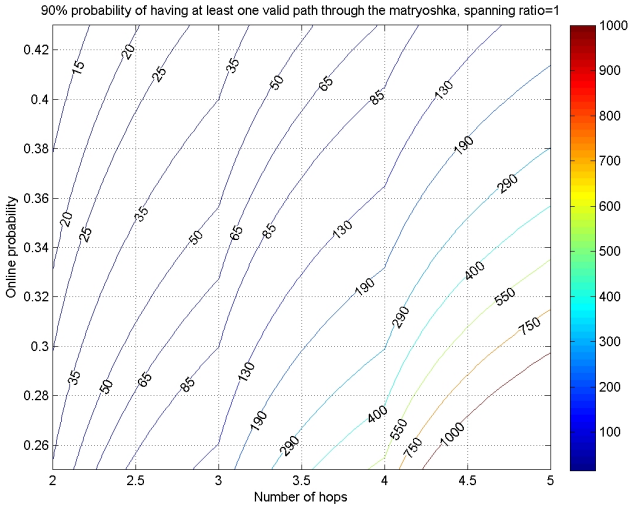


**Figure 3. Access data of a user - span=1.**

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

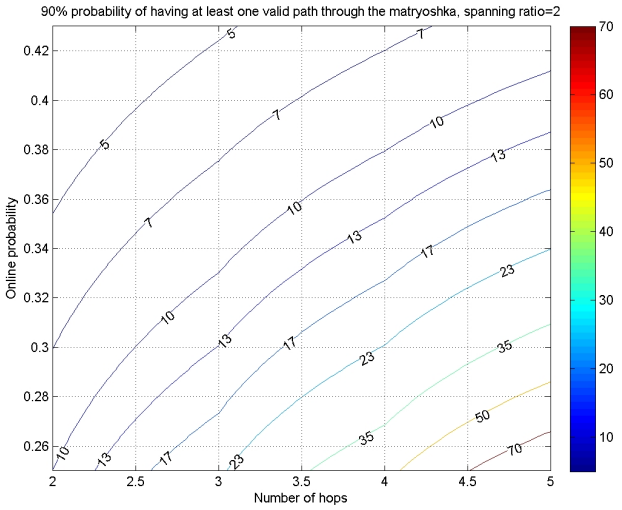


**Figure 4. Access data of a user - span=2.**

**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 $\Omega$ 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 $\omega_j$'s friend list and be able to determine which one of $\omega_j$'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 $\lambda_j$ in the inner shell $\Lambda$ of $\mathcal{A}$'s matryoshka is

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

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

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

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 $p_\Lambda$ 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 [10] due to its short response time. According to recent studies [14] 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.

Based on the results of [14] we did a monte carlo simulation and the overall data lookup time $T_{dr}$ is expected to take 9.2 seconds, with a median of 8 seconds and 90% of the lookups taking less than 13.5 seconds.
Following the proposed changes to KAD this latency could be significantly reduced. Note that the full retrieval delay will only be necessary for the first time a user accesses y's data, and the KAD lookup could be avoided for later lookups, if the known entry points to y's matryoshka are online. The expected latency for data retrieval in this case is 2.7 seconds, with a median of 1.7 seconds, and it would be less than 5.4 seconds for 90% of all requests, without taking into account that the social proximity can correspond to the geographical one.

## 4 Related work

We see Safebook to be related to work on three different fields: P2P, OSN, privacy (see fig 5). While previous work usually focuses on one of the intersection of two of these fields, Safebook, to the best of our knowledge, is the first work integrating all three. In this section we present an overview of that previous work.

Current OSNs offer privacy protection by hiding particular sets of personal data, but they lack protection against access to the central storage due to their centralized architecture. In order to address this issue, Safebook preserves the user's privacy by following a decentralized approach.

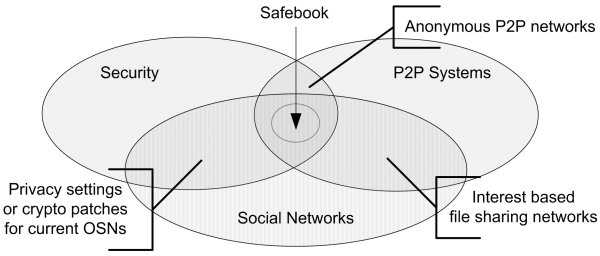


**Figure 5. Work in secure OSN P2P systems.**

Privacy preserving data exchange is offered in a distributed fashion by a series of anonymizing Peer to Peer systems. However, differently than Safebook and, generally speaking, OSN's purposes, they are primarly created for filesharing applications and other kinds of asynchronous exchanges.

Several anonymized P2P systems improve their performance by creating groups of interest. These groups are defined by relations that are established during resource lookups. These relations do not exist in real life, so they are different to edges in a social network as friendship links in Safebook.

**Privacy preserving OSN** Not much work has been done on security and privacy issues in OSN.

The work presented in [8] raises a series of privacy threats related to OSN, varying from identity theft, private data collection, espionage, SNS spam, face recognition and so on. In Safebook users can't cheat about their identity, which is certified by a trusted identification system.

NOYB [7] breaks up private members' profile in atoms, then swaps these atoms with false ones according to public available dictionaries, possibly distributed in a peer to peer fashion. The swap process involves the use of symmetric keys provided to friend contacts out of band, in order to let them retrieve correct atoms. Safebook achieves the same fine grained access control on users' data with group key cryptography techniques. This avoids the use of additional structures like dictionaries, whose access is time consuming and whose size increases over time.

**Privacy preserving P2P systems** A lot of work has been done to guarantee security and privacy in P2P networks. A survey of anonymous P2P networks is presented in [13] and [4]. anonbib[5] additionally gives a good overview of existing approaches.

Like Safebook, Freenet [9] 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 communi-

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

cation 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 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 [1] 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.

**P2P and OSN** 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 [3] and Bittella [12] 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 [3] and [12], 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 Conclusion and Future Work

In this paper we presented a preliminary feasibility study of a new privacy preserving OSN[5]. This new architecture focuses both on decentralization in order to avoid a central omniscient entity and on leveraging trust relationships from the social network to enforce privacy and cooperation enforcement in the OSN system. Decentralization is provided thanks to a structured peer to peer substrate, while privacy and cooperation enforcement are assured through to the matryoshkas, concentric structures crossed by paths providing hop-by-hop trust anonymizing each OSN member and guaranteeing access to the data of users.

It is sufficient for each user to set from three to four shells in its matryoshka and have up to 23 of his trusted contacts storing his data in order for his data to be accessible with 90% probability. In this case the overall data lookup delay can be estimated to be below 13.5 seconds for the 90% of all the initial requests without using any prefetching or other methods of mitigating lookup delays.

We are currently building simulation models to run extensive studies on the performance of Safebook, especially with respect to different numbers of shells in the matryoshka and different P2P substrates. We are implementing a preliminary client of Safebook at the same time to show its feasibility and performance in a real world environment.

## References


[1] K. Bennett and C. Grotho. Gap - practical anonymous networking. In *Privacy Enhancing Technologies workshop. Springer-Verlag, LNCS 2760*, pages 141–160, 2003.

[2] 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.

[3] V. Carchiolo, M. Malgeri, G. Mangioni, and V. Nicosia. Prosa: P2p resource organisation by social acquaintances. pages 135–142.

[4] T. Chothia and K. Chatzikokolakis. A survey of anonymous peer-to-peer file-sharing. In *Network-Centric Ubiquitous Systems*, pages 744–755. Springer.

[5] L. A. Cutillo, R. Molva, and T. Strufe. Privacy preserving social networking through decentralization. In *Wireless On-demand Network Systems and Services*, Feb 2009.

[6] 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.

[7] S. Guha, K. Tang, and P. Francis. Noyb: privacy in online social networks. In *Online social networks*, pages 49–54, 2008.

[8] G. Hogben. Security issues and recommendations for online social networks. Technical Report 1, 2007.

[9] 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.

[10] 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.

[11] A. Poller. Privatsphärenschutz in Soziale-Netzwerke-Plattformen. Fraunhofer SIT Survey, `www.sit.fraunhofer.de`, 2008.

[12] 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.

[13] 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.

[14] M. Steiner, D. Carra, and E. W. Biersack. Faster content access in KAD. In *Peer-to-Peer Computing*, Sep 2008.