Gemstone: Empowering Decentralized Social Networking with High Data Availability

Florian Tegeler, David Koll and Xiaoming Fu
Institute of Computer Science
University of Goettingen, Germany
Email: [tegeler|koll|fu]@cs.uni-goettingen.de

***Abstract*—Social networking platforms such as Facebook, MySpace, and Twitter have seen a significant increase in user population and user provided information. However, users are increasingly concerned about identity and data privacy since information is controlled by single companies. To address this issue researchers investigated alternative solutions, where the users' data, e.g. profile information, comments and messages, is stored at user-controlled nodes. Although these solutions provide a plausible means for avoiding privacy leaking in central instances, they raise a new challenge to design a cost-effective storage replica scheme which ensures a high data availability even when some users are offline.**

In this paper we present Gemstone, a social network platform where the data replication scheme leverages a learning mechanism based on social relationships, online patterns of peers and user experiences. Our preliminary evaluation shows that compared to related works, it achieves higher data availability while requiring a smaller number of data replicas.

I. Introduction

The tremendous information aggregation at the operators of growing online social networking (OSN) platforms raises a privacy concern: Providers obtain a deep insight into a user's social relationships, personal opinions and economical or political preferences and may base their business model on exploiting such information. In Facebook's Beacon application (2007-2009), partner websites of Facebook were - initially without consent of the users - enabled to use personal data of users in targeted advertisements. Moreover, status information (e.g., shopping activities) about the user were transmitted to the Facebook central servers. For the professional social networking platform Xing, Wondracek et al. [1] show that it is possible to de-anonymize users at independent websites. Balduzzi et al. [2] show that linking profiles of different social networks and using the common friend-finder service allows to detect hidden profiles and reveal a user's usually not public e-mail address.

The combination of data encryption and decentralization is a key element to ensure data protection for the following reasons:

- *Encryption*, which is independent from the social networking site provider, enables users to protect their data from unauthorized acces. Additionally, sophisticated mechanisms such as Attribute Based Encryption (ABE) [3] allow users to accurately define access rights to their data. This access can be granted, e.g., based on social criteria such as being a member of a certain group or a friend.
- *Decentralization* of data storage prevents a single entity to observe profile access or - in general - data flows inside the social network. Such data access could, e.g., be used to derive patterns and retrieve friendship information or identities although users configured such to be private. Additionally – in a decentralized social network – a data leak at a single storage location does not leak a large portion of the network's data, which otherwise could be analyzed and potentially decrypted.

Whereas it is possible to address the encryption issue, e.g., using ABE [4], a cost effective data distribution that allows high data availability with a minimal number of replicas is challenging. This is especially the case, as traditional data storage theory (see [5], [6], [7]) based on observations of file sharing and data center distribution can not be applied for the following reasons:

- In OSNs, users' online patterns show high activity peaks with larger gaps of offline time [8]. In contrast, the often studied file sharing networks are characterized by long durations of online activity, typically spanning multiple hours up to days.
- Content in social networking platforms is often uploaded from mobile devices which may be disconnected most of the time, e.g., for the sake of saving energy.
- The inherent relation between the participants may impose implications on storage incentives in P2P like organized decentralized storage: It can, for example, be assumed that a user prefers to store the data of his friends to that of a stranger.

Additionally, a decentralization approach encounters the critical challenge to ensure that a user's profile information is highly available even during the user's offline periods. This can be achieved in the following basic means:

1) Data control and storage can be distributed to a limited number of permanently online storage locations, which are shared by multiple users. In such scenarios as for example realized in Diaspora[1], 100% data availability is achieved and storage might either be altruistically provided or based on an economic incentive such as paying users. In Diaspora these storage locations are called pods

[1] http://blog.joindiaspora.com/what-is-diaspora.html

and are currently altruistically contributed by universities or private persons. However, considering more than 500 million active users currently on Facebook[2], altruistic provisioning seems rather unlikely in a large-scale social network.

2) Each user might be able to provide a permanently available storage space for its own profile. This solution, e.g. fostered in Persona [4], provides 100% data availability as well but requires casual users to be technically able to provide data storage such as their own webspace. This problem might be circumvented by economically motivated storage providers, resulting in higher costs for the user.
3) Assuming no permanently online storage is available, nodes can cooperate and provide temporarily available storage to each other. A major challenge in this scenario is to achieve 100% data availability, although our results for such a system indicate a relatively good approximation. However, taking the advantage of mutual cooperation of nodes and flexible data storage locations into account allows users to be independent of altruistic servers that may disengage after some time. Additionally, the network can operate without additional costs for the user as every participant is contributing data storage.

The latter approach was also proposed by Buchegger et al. in PeerSoN [9] and Cutillo et al. in Safebook [10]. PeerSoN is a decentralized social network utilizing an optimized selection algorithm based on cliques with mutual storage agreements [11]. The algorithm assumes that node online times are known and follow similar patterns as file sharing. The authors modeled the online behavior according to a file sharing study [12] and simulated 70% of the nodes with an online probability of $p \geq 0.75$. For such scenarios, data availability rates of above 99% were estimated.

Safebook [10] is a P2P-based social overlay in which each node is accessible through so-called shells, a logical grouping of socially related nodes that form a Matryoshka doll like structure. Profile data is mirrored at a subset of a node's direct contacts which form the innermost shell. The data retrieval technique is similar to onion routing [13] and requires traversal of the shells along a path of simultaneously online nodes that befriend each other. As a consequence, for an increasing number of shells there is the trade-off between increased privacy (in this case especially anonymity) and performance, as each shell requires a significant increase in mirroring nodes. For Safebook, reasonable data availability of 90% was achieved using 13 (3 shells) or 23 (4 shells) replica nodes.

A. Our Contributions

In this paper, our contributions are twofold:

- Firstly, we propose Gemstone[3], a generic P2P social networking system that functions as a middleware to support different OSN applications. Gemstone assists these applications by proving a shared social graph, serving profile information and handling messages delivery to peers. Gemstone is completely decentralized and protects the user's privacy by encrypting all data using ABE.
- Secondly, we present a first approach towards a sophisticated data storage solution based on data replication. The mechanism ensures high data availability even during a peers offline time and exploits the potential availability of (altruistically provided) permanent storage but does not rely on it. This hybrid approach achieves high data availability using only a minimal number of mirrors. It leverages social relations, online times and observed past node behavior to select suited storage peers. By taking the advantage of mutual cooperation of nodes and flexible data storage locations into account, our approach allows users to be independent of altruistic servers that may disengage after some time.

The remainder of the paper is organized as follows: In Section II we present Gemstone, our social networking software. Gemstone implements our novel data storage scheme which is introduced in Section III. We evaluate our approach in Section IV and compare it with related work. Security aspects are considered in Section V. Finally, Section VI concludes the paper.

II. Gemstone

Gemstone is a generic P2P social networking overlay that resides as a middleware between social applications and the networking stack. Figure 1 illustrates the current structure. A user's data such as the shared profile, the social graph and messages received from other nodes are stored at the peer itself. Additionally, the encrypted data is replicated among other nodes to ensure data availability during the peers offline times. These, so-called Data Holding Agents (DHAs) are selected following a sophisticated algorithm which we will present in detail in Section III. One of Gemstone's inherent properties is the synchronization of social data among all participating applications and multiple devices of a user. Hereby, we enable multiple concurrent applications on top of Gemstone - for example a decentralized Facebook and a decentralized MySpace - to share a common social graph.

The overlay is formed using a Distributed Hash Table (DHT) to store routable identifiers of Gemstone participants, profile locations and DHAs of peers. During its online time, a node serves its user's profile and ensures that synchronized replicas are available at the chosen DHAs. In its absence, intelligently chosen DHAs ensure a high data availability and accept messages on the profile owner's behalf.

A. Data Exchange

For data exchange between Gemstone nodes we use simple data containers as shown in Figure 2 that implement a transparent message concept. Thus, the platform functions as a (de)multiplexing unit that distributes the data objects to the correct applications according to a globally unique application

[2]http://www.facebook.com/press/info.php?statistics

[3]The source code is available at http://gemstone.informatik.uni-goettingen.de

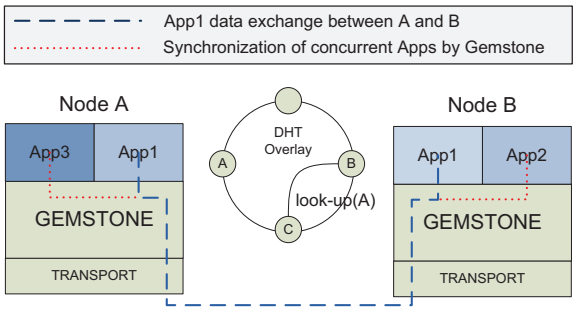


Fig. 1: System Overview

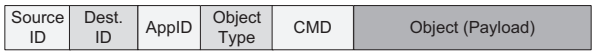


Fig. 2: Object Format

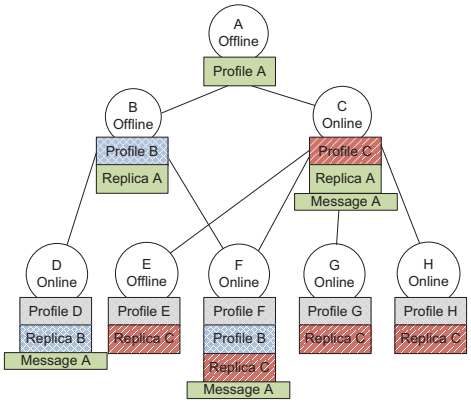


Fig. 3: Message Replication

identifier (AppID). The object type in combination with the command field (CMD) can be used to differ between application objects with an opaque object payload and middleware control types such as messages, data storage or social graph updates. The object type hereby specifies whether the object is a Gemstone object or an external application object and the command contains the details action that should be performed on the object's payload. Thereby, applications can easily send any type of data from one application to another but can also manipulate the middleware by using the appropriate object-types. Effectively, profiles, friendship lists and entries such as comments or wall entries are all stored as messages dedicated to a given user.

B. Storing and Retrieving Data

Data is stored in form of encrypted user profiles and messages to users, which can contain arbitrary contents. In our current design, an online user directly receives messages and presents its profile, whereas offline users (e.g., A in Figure 3) require assistance by their replica nodes (e.g., B and C in Figure 3) that receive the messages and store them to be collected by the returning online user. In Figure 3, DHA B itself is offline again, so that messages destined for A have to be passed on to B's mirrors D and F. B retrieves these messages upon returning online according to Algorithm 1 (we will explain p^{ue} in the following Section). Hereby, all online DHAs always present the most recent profile version. Also, a user's profile updated via messages from different devices remains automatically synchronized.

C. Confidentiality of Data

The confidentially of all privacy relevant user information is a key element in Gemstone. Therefore, we intend to encrypt all data by using Attribute Based Encryption (ABE) [3]. In ABE, the symmetric key for encrypted content is protected by an Access Structure, which is defined by a combination of attributes so that only requesters holding the correct attribute key can decrypt it. The user can then assign such attribute keys – e.g., containing the attribute 'friend' – to a group (e.g., all family members) or individual users explicitly. This allows each user to grant fine grained access to his confidential data, as it can not be accessed by other entities than those holding the corresponding decrypting key. Moreover, the impact of the overhead introduced by the encryption routines is limited, even on mobile devices [4]. Therefore, ABE provides data confidentiality alongside finely grained access control without introducing much overhead.

Algorithm 1 Data Retrieval for a returning node n

look-up DHA_n
for all $i \in DHA_n$ **do**
  if i is online **then**
    n retrieves profile data
    n retrieves messages
    for all j where $n \in DHA_j$ **do**
      n retrieves messages destined to j
    end for
    increase p_i^{ue}
  else
    decrease p_i^{ue}
  end if
end for

D. Deployment Considerations

We envision Gemstone to be incrementally deployed by integrating into existing online social networks such as Facebook or MySpace: A wrapper application on top of Gemstone uses the Facebook API[4] to extract all relevant data concerning the basic profile, the friendship graph and the group memberships. Additional data from Facebook has to be handled inside the application. The Facebook social graph is injected into Gemstone and mapped to Gemstone users using the application interface. Afterwards it is available for all social based applications interfacing Gemstone. If desired, the wrapper application can update the centralized Facebook server over all changes made in the Gemstone network. Hereby, a synchronized copy of the

[4]http://developers.facebook.com/

original social data is maintained in Gemstone the user can merge the social networks from Facebook, Flickr, MySpace, etc. in Gemstone and slowly migrate away from centralized platforms.

III. Replica Selection

A user's social networking data can be retrieved from mirroring DHA nodes during times in which the user himself is offline. For high data availability it is therefore crucial, to select DHAs that are frequently online, willing to mirror data for the user and serve the profile to requesting nodes. In order to not overwhelm the resources of individual DHAs, as few peers as possible are chosen to reach high data availability. Hereby, the average number of profiles each peer has to mirror is minimized. In a first approach towards a sophisticated solution we propose the following: Each peer locally selects its DHAs based on an estimated likelihood to successfully preserve data during the data originators offline time. This likelihood p_i is calculated for each candidate mirror node i based on three aspects:

1) The **online time** $p^{ot} \in [0,1]$, which represents the average online probability. This value is measured and provided by the candidate nodes themselves during normal communication such as DHT lookups, profile pulling or message forwarding.
2) The **social relation** $p^{sr} \in \{0,1\}$, which is currently an absolute measure of either being a friend or not. This information is directly accessible using the local social graph.
3) A personal **user experience** $p^{ue} \in [0,1]$. Each time, a peer i goes offline and returns online, it checks its profile availability av_i by verifying that the nodes which were supposed to mirror its profile are online and serving all data. If a peer $j \in$ DHA is online and can present the profile, a positive experience is logged, if j is offline or has dropped the profile, the event is unsatisfactory as shown in Algorithm 1. This way we implicitly consider permanent storages as p^{ue} is maximized for such peers.

As choosing the optimal nodes with a fixed number of n mirrors would reduce to an NP-complete (Knapsack) problem, we chose a sub-optimal solution, where each node orders its known candidate nodes by p_i and selects the most valuable nodes until the estimated error probability is very low:

$$\prod_{i=1}^{n}(1-p_i) < 0.01 \tag{1}$$

Following this selection process, the number of mirror nodes selected by different nodes differs as the set of known potential replica nodes is locally created at each node. This set continuously grows as new nodes are discovered, for example during DHT traversal to look up a friend's profile. To increase system scalability by preventing infinite growth of these sets, each node assigns fixed Time-To-Live (TTL) values to its candidate set peers. The TTL is reduced each time a node is not chosen as a DHA and on TTL reaching zero the peer is removed from the candidate set. Note that socially well connected ($p^{sr} = 1$) persistent storage is chosen frequently, as its p^{ot} will be near 1. If such storage space fills up and profiles have to be dropped, the user experience value p^{ue} will lower and the peer is chosen less frequently. As a result, persistent storages are therefore highly utilized by socially related peers and supplement fluctuant user provided storage.

IV. Evaluation

We analyzed our replica node selection algorithm via simulation and used four strategies to investigate the effects of the different input aspects:

1) **Random**: A basic strategy that selects known nodes randomly and ignores the aforementioned likelihood ordering.
2) **Online time**: The calculation of p_i is solely based on the online time information p_i^{ot} provided by the candidate DHAs.
3) **User experience**: Additional to the online time information, the personal user experience p_i^{ue} is included in calculating p_i.
4) **Social relation**: The user experience strategy is extended by inclusion of the social relationship p_i^{sr} to a candidate storage user. We hereby assume that nodes have an incentive to store data for their friends and behave well to their socially related peers, which results in a modified drop policy and less attacks at the replica node side.

Furthermore we simulated our approach under the same online time assumptions as of related work and compared the results in terms of overall data availability and the average number of required replicas.

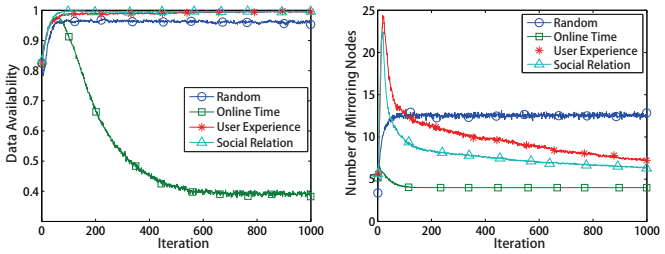
A. Simulator Setup

Social relations between users follow the typical power-law distributed scale-free small-world network characteristic [14], [15]. We used five different social graphs of 1,000 to 10,000 users. On average, each user in our simulation has 130 social relations, which is the current average of a Facebook user's number of friends[5]. There are however users without friends in the resulting graph, which represent new users joining the system.

Node online times are also power-law distributed following recent studies [8], [16]. This power-law assumption results in two thirds of the users being online less than 20% of the time in our simulation. Furthermore, there is no correlation between the number of social links a user has and his online time: A user with a large number of relations is not automatically more available than a user with less relations. In fact, real OSN users become less active as their number of relations increases in an OSN [8].

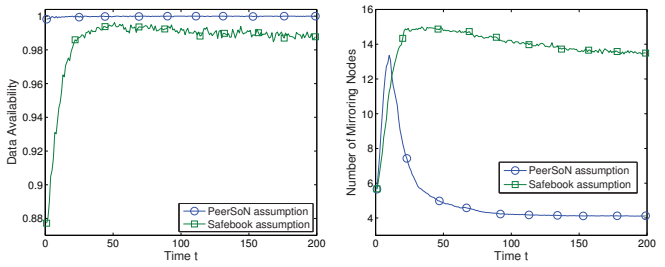
Storage space to mirror profiles for other nodes is limited at each node and follows a normal distribution with a mean value of 50 profiles. This is a relatively low number as data such as wall entries, friendship information, group data and

[5]http://www.facebook.com/press/info.php?statistics



(a) Data availability by strategy (b) Number of DHAs by strategy

Fig. 4: Simulation Results



(a) Data availability under PeerSoN and Safebook assumptions (b) Number of DHAs under PeerSoN and Safebook assumptions

Fig. 5: Simulation results under different assumptions

messages is primarily text information. If the storage space is completely consumed, nodes drop profiles following a social drop policy: peers preferably drop socially unrelated profiles.

Learning rates are used to model the growth of the set of candidate nodes. In our simulation, we operated conservatively assuming nodes only discover at most $log(n)$ new nodes per iteration (one DHT-lookup), depending on their activity factor. This factor represents the node activity in the network and is modeled gaussian distributed.

B. Results on Strategy Efficiency

We evaluate the average data availability for each strategy. In each iteration, we check the availability for each node's profile (1 if at least one peer serves the profile, 0 if the profile is unavailable) and calculate the average. This averaged data availability is shown in Figure 4a for each strategy. The corresponding average number of DHAs can be seen in Figure 4b. For the random selection strategy data availability of about 96% is achieved, but with a relatively high cost of 12 replica nodes. When selecting based solely on node online times, data availability is initially $\approx 97\%$, but drops rapidly after around 80 iterations to low 40% afterwards. This overloading-effect is based on overflowing storage space at the nodes having the most online time. As online time knowledge propagates throughout the system an increasing number of nodes stores its profiles at such highly available peers. In contrast to this, selection based on user experience achieves high availability of above 99% which is maintained stably throughout the whole simulation. Additionally, considering social relationships leads to a slightly quicker convergence and higher data availability in the early stages of the simulation.

We observe that for all scenarios data availability is around 80% at the beginning. This is caused by the power-law distribution of social relations in our graphs, where many nodes only maintain very few social relations or even join the system without any relation. Initially, relatively inactive and insufficiently socially connected nodes have problems to locate potential DHAs. Due to the learning effects in the network, data availability improves quickly, even with our very conservative learning model.

C. Comparison with Related Work

In a first comparison to related work, simulating our approach under the online time assumptions of Safebook and PeerSoN respectively results in increased data availability, while using less replica nodes as shown in Figures 5a and 5b. For the Safebook distribution, data availability of above 98% was achieved requiring around 14 replicas. Compared to that, Safebook achieves 90% availability by using 13-23 replicas (for 3 to 4 shells). For the same node availability distribution as in PeerSoN, our solution works with practically zero losses and only four mirroring nodes on average respectively. In comparison, PeerSoN achieves data availability of $\leq$90% up to $\approx$ 100%, depending on a node's own availability, by using clique sizes of 5. In this system, data of nodes with little online time achieve the least availability. On the contrary, our system does not distinguish between node availabilities. Additionally, our system is fair with regards to nodes with only few social relations: In Safebook, a node with very few contacts might be unable to build its Matryoshka properly, as it may not be able to provide enough nodes for the innermost shell. Also, the less available a user's friends are, the less available a path through the shells and therefore the user's data is. In contrast to this, our solution only prefers socially related nodes as mirrors over unrelated nodes if they have a comparable user experience value or high online times, thereby neutralizing the dependency on the quality of a node's social relations.

V. Security

Gemstone enforces encryption and decentralization in social networks which strengthens data protection. However, a variety of attack scenarios needs to be considered:

A. Crawling

Crawling is hard as friendship lists in Gemstone are encrypted. Hence, knowing a set of users does not allow to systematically crawl their social graph. A potential security risk for crawling might be imposed by the DHT, which could be traversed to obtain data. Nevertheless, as the data is encrypted, an attacker's insight can be assumed to be limited.

B. Data Aggregation

Our data storage algorithm prevents information aggregation at a single provider (or node) and thereby thwarts data misuse for economical - as in Facebook's Beacon - or political reasons. By biasing the selection to favor social links, data is

preferably stored at friends, which prevents data aggregation even at highly available nodes. Nevertheless, an attacker might try to obtain many friendship relations, either by automatically sending out a high amount of requests or by imitating real user's profiles that help lurking users into friend request acceptance. According to studies by Bilge et al. [17] 20% of social network users accept friendship request from completely unknown profiles. Hereby the attacker is mistakenly classified as a friend, which grants access to parts of the user's data and increases the likelihood to be chosen as a DHA. Although such kinds of attacks are platform-independent and exist in OSNs in general, our system mitigates the impacts by using ABE. It is unlikely for a user to assign a high trust with a completely unknown profile. Therfore, the attacker will not obtain all attributes to decrypt all - especially the most sensitive - personal data of the user.

C. DHT attacks

A sophisticated attacker might analyze DHT control traffic to obtain statistical data on users. Hence, we believe this attack to be of minimal impact as all data and identities are encrypted and thereby protected from indepth analysis. Nevertheless, tampering with the DHT or rejection to forward legitimate nodes lookup requests might impact our scheme.

D. Attacks at the Selection Scheme

We are also aware that biasing the node selection scheme on online time information provided by candidate nodes itself introduces new attack vectors to the system: Attackers might try to decrease the overall system stability or obtain data by signaling a 100% online availability and receive a large number of replicas. Furthermore, attackers can perform a profile hiding attack by selectively signalling high availability times to a specific node that finally stores most of its replicas at attacker nodes. In our selection scheme, such attacks are mitigated by relying on the personal user experience with each node as well. Moreover, we are currently considering a stronger inclusion of learning mechanisms to increase the level of control to further reduce the possibilities for this kind of attacks. This can be achieved by, e.g., verifying whether announced online times comply to observed online times, and obtaining recommendations for suited replica peers from trusted peers, such as social contacts.

VI. Conclusion and Future Work

This paper presented Gemstone, a middleware for social networking applications that increases data privacy by enforcing encryption and decentralization. For the latter, our hybrid replica selection algorithm leverages nodes' online time information, social relations and personal experiences. A preliminary evaluation of our approach shows an increase in data availability and a reduction of the number of replica nodes in comparison with related work. Overall, our system allows generic decentralized social networking while providing high data availability with a low number of replicas. In our future work, we will elaborate our replica selcection algorithm to close the still existing gap to a centralized system in terms of data availability. Moreover, a more thorough evaluation of the algorithm is required. Finally, we will address the specific characteristics of mobile devices that communicate using Gemstone by designing special "mobility service nodes". These resourceful nodes are envisioned to serve the mobile nodes as a kind of dynamic cloud that effectively will reduce DHT churn and lower the energy and bandwidth consumption of the mobile devices.

References

[1] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A Practical Attack to De-anonymize Social Network Users," in *IEEE S&P '10*, May 2010.

[2] M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel, "Abusing social networks for automated user profiling," in *Proceedings of the 13th international conference on Recent advances in intrusion detection*, ser. RAID'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 422–441. [Online]. Available: http://portal.acm.org/citation.cfm?id=1894166.1894195

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in *IEEE SP '07*, 2007.

[4] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: An Online Social Network with User-defined Privacy," in *SIGCOMM '09*. ACM, 2009.

[5] J. Kangasharju, K. Ross, and D. Turner, "Optimizing File Availability in Peer-to-Peer Content Distribution," in *INFOCOM '07*, May 2007.

[6] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. Ramakrishnan, "Optimal Content Placement for a Large-Scale VoD System," in *CoNEXT '10*. ACM, 2010.

[7] P. Rodriguez, S.-M. Tan, and C. Gkantsidis, "On the Feasibility of Commercial, Legal P2P Content Distribution," *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 75–78, January 2006.

[8] L. Gyarmati and T. Trinh, "Measuring User Behavior in Online Social Networks," *IEEE Network*, vol. 24, no. 5, pp. 26 –31, September 2010.

[9] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta, "PeerSoN: P2P Social Networking: Early Experiences and Insights," in *SNS '09*. ACM, 2009.

[10] L. Cutillo, R. Molva, and T. Strufe, "Safebook: A Privacy-preserving Online Social Network Leveraging on Real-life Trust," *Com. Mag., IEEE*, vol. 47, no. 12, pp. 94–101, 2009.

[11] K. Rzadca, A. Datta, and S. Buchegger, "Replica Placement in P2P Storage: Complexity and Game Theoretic Analyses," in *IEEE ICDCS '10*, June 2010.

[12] S. Bernard and F. Le Fessant, "Optimizing Peer-to-Peer Backup Using Lifetime Estimations," in *EDBT/ICDT Workshops '09*. ACM, 2009.

[13] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," in *13th USENIX Security Symposium*, August 2004.

[14] M. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM REVIEW*, vol. 45, pp. 167–256, 2003.

[15] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," in *SIGCOMM IMC '07*. ACM, 2007.

[16] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida, "Characterizing User Behavior in Online Social Networks," in *IMC '09*. ACM, 2009.

[17] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, "All your contacts are belong to us: automated identity theft attacks on social networks," in *WWW '09*. ACM, 2009.