SocialGate: Managing Large-Scale Social Data on Home Gateways

David Koll, Dieter Lechler and Xiaoming Fu

University of Goettingen, Germany
{dkoll,xfu}@cs.uni-goettingen.de, dieter.lechler@stud.uni-goettingen.de

***Abstract*—Today, Online Social Networks (OSNs) are ubiquitous means of communication. In order to prevent the misuse of personal user data by OSN providers, various research efforts have produced a multitude of approaches to *decentralize* OSNs in the past decade. The most critical challenge for these systems is to replace the infrastructure of centralized OSNs. That is, they need to handle the large amounts of data uploaded by users on one end, and requests towards that data on the other end. Typically, existing approaches instrumentalize cloud facilities or user devices for this task. Unfortunately, they introduce either a monetary cost for users or have limited success in making data highly available. In this work we propose *SocialGate*, the first prototype that makes use of home routers of users as the infrastructure backbone of the OSN to avoid these shortcomings. Measurements and experiments based on real-world data support the feasibility and practicability of our approach.**

I. Introduction

Online Social Network (OSN) providers such as Facebook or Twitter store massive amounts of data at their premises. In particular, they typically have insight into vast amounts of private data uploaded by their users, which yields unprecedented access to sensitive information such as political views, personal preferences or the contents of private messages. On various occasions and regardless of recurring lawsuits and critiques on their behavior, providers have shown to exploit and analyze this information without user consent [1], [2].

In order to mitigate these privacy breaches, over the past decade researchers have developed a wide range of ideas to decentralize OSNs [3]. The common philosophy and goal of these approaches is to remove the central data repository that is at the core of current OSNs, and instead to return the control over user data back to the users [4].

While the approaches to facilitate user control over data are specific to each particular solution, this goal also results in one major challenge: in the absence of a central data store, large-scale social data has to be managed such that it is available for interested and eligible users. Hence, the critical challenge of each DOSNs is to build a surrogate for the infrastructure previously maintained by the OSN provider.

Until now, research has focused on two major concepts to build this surrogate. One direction is to facilitate data storage at distributed, but provider-owned storage locations (such as Dropbox or paid web servers) in encrypted form (e.g., [5]–[10]). This allows for (i) data storage at facilities with large capacity, (ii) an efficient infrastructure surrogate with little storage and communication overhead, (iii) retrieval of user data with high speed from dedicated servers, and (iv) high availability of data due to the use of cloud facilities. At the same time, this approach usually incurs some sort of subscription-based monetary cost on the user and again places considerable amounts of data at the premises of a service provider. This provider can further at all times alter the terms of use, increase subscription fees, analyze usage patterns of its users or even stop the service completely.

Orthogonal to this idea, a different range of solutions lets users contribute their own devices to store all OSN data in a user-provided overlay (e.g., [11]–[15]). This approach can usually (i) operate free-of-charge and (ii) eliminate any centralized involvement. However, it struggles to achieve stable data availability and low replication overhead, as user devices are not as reliable as server or cloud resources. They typically experience high churn, which renders making data highly available for retrieval a challenging task.

In this work we introduce *SocialGate*, a novel solution that combines the advantages of both directions while avoiding their downsides. In particular, SocialGate eliminates any centralized involvement and at the same time is able to make data highly available with only marginal overhead. It further facilitates physical control of users over their own data and only requires a small one-time investment as opposed to monthly fees.

In a nutshell, SocialGate creates a Distributed Hash Table (DHT) overlay to interconnect home gateways, which often come with server-like availability and connectivity. These devices then store user data that is encrypted with Attribute Based Encryption [16] to ensure that only eligible network participants can access a user's data. Those eligible users can then directly request data from the respective home gateways without any involvement of a central entity. While previous approaches required to deploy at least four (and often many more) copies of user data across the network to make data reasonably well available, SocialGate only deploys a single

978-1-5090-6501-1/17/$31.00

replica in the network. Based on real-world measurements of availability data, our evaluation shows that this is sufficient to make data available for more than 99% of the time. Adding a second copy can increase data availability to more than "three nines" (99.97%), but comes at the expense of introducing synchronization overhead among replicas.

We provide the first generic, publicly available implementation (in Python) of a DOSN that can run on both home gateways and—for users without access to a home gateway—standard operating systems. We deploy SocialGate in a small testbed of home gateways and user devices and find in real-world measurements that the latencies induced by our approach are well within reasonable bounds.

The remainder of this paper is structured as follows: In Section II we briefly discuss related work. Section III introduces the SocialGate system. In Sections IV and V we describe our implementation and evaluate SocialGate, respectively. Finally, we conclude the paper in Section VII.

II. Related Work

Approaches towards decentralizing OSNs are plentiful. In [3], we present a survey on the most recent advances on DOSN and discuss the advantages and disadvantages of over 20 solutions. In general, these solutions can be generalized to three broad categories of solutions.

A. Server-based Solutions

Server-based solutions (e.g., [5]–[10]) keep data storage and management at central data repositories. For instance, users store their data on services like Dropbox [9]. The key contribution of these solutions is that they offer a way for users to encrypt their data at these central services, making data analysis more difficult for the provider. However, providers can still track access to data, try to de-anonymize users, introduce or increase storage fees, change their terms-of-use or even stop providing the storage service at will [3].

B. User-cooperation Solutions

To overcome these issues, a second branch of solutions lets users provide their devices (computers, laptops or even smart phones on one hand, but also personally owned servers on the other hand) to cooperatively replace the data infrastructure of the provider (e.g., [11]–[15]). This completely eliminates the involvement of a central provider, but comes with downsides in data handling. All cooperation-based systems have difficulties to make data highly available (due to the churn in user devices), and are typically not entirely successful in doing so at the expense of increased management and communication overhead [3].

C. Hybrid Solutions

Further, hybrid solutions (e.g. [17]–[19]) aim at combining the advantages of both solutions, but unfortunately often end up with also combining the respective disadvantages [3]. For instance, a solution that involves a cloud provider for setting up a directory service and user devices for storage of data suffers from both potential provider misbehavior and churn of user devices [3].

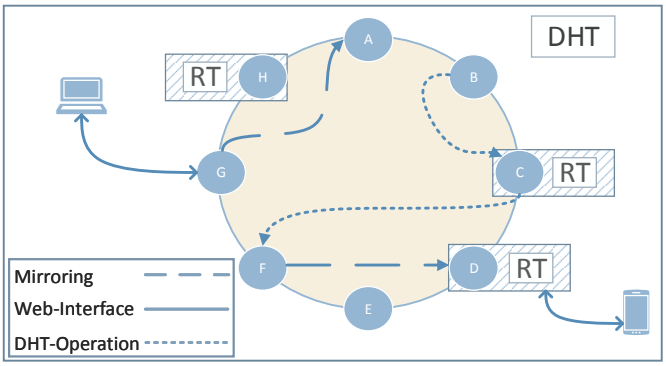

Fig. 1. SocialGate Overview

D. Solutions for Embedded Devices

Finally, there exist a few approaches that target embedded devices as data storage locations. In [20], Marcon et al. discuss the feasibility of implementing social networking from home, but lack a system description and implementation. Recently, the Databox project has started to provide home storage for data [21], but does currently not target OSN applications. Finally, Mastodon[1] can potentially be run on Raspberry Pi, but is limited to microblogging. It further requires (i) buying a Raspberry Pi, while home gateways are usually present in most homes already, and (ii) executing a sophisticated installation routine[2], which is not easy to follow for general OSN users.

III. SocialGate

We now introduce *SocialGate*, our DOSN solution that is built on home gateways. We first introduce the system architecture and then discuss challenges that arise when employing gateways as storage devices.

A. System Overview

SocialGate exploits a distributed hash table (DHT) as an overlay for storing a directory of DOSN users as shown in Figure 1. Users can perform lookups for other users in the DHT, and subsequently directly communicate with each other to request data that is maintained at user devices. The nodes participating in the DHT are primarily aimed to be home gateways (or *routers*, RT in Figure 1), where available. This has a series of advantages:

- First, it keeps data not only logically, but even physically under the control of its owner. Home gateways are—as their name suggests—usually deployed in the home of users, while previous approaches often stored data in remote locations. This also implies that, for instance, an organization can provide multiple nodes such that social data is only stored within the boundaries of the organization.

[1]https://joinmastodon.org/
[2]https://hub.docker.com/r/gilir/rpi-mastodon/

- Second, it benefits from the high availability of home routers to achieve stable data availability with little overhead. These two metrics are key for the practicability of a DOSN solution. Only in high availability DOSNs the user experience can be similar to current, centralized OSNs, and high-overhead solutions will unnecessarily burden the infrastructure substrate, i.e., user devices.
- Third, it offers good performance due to typically high bandwidths at land lines. Whereas previous cooperation approaches often pushed data to smart phones of users with possibly limited bandwidth, SocialGate is anchored in an environment that can easily answer even multiple requests per second.

1) SocialGate DHT Overlay: The DHT allows to efficiently publish and lookup contact information for each user. In particular, every user can publish relevant information in a directory entry in the DHT. This entry typically contains the user's name and her unique SocialGate ID, the IP address of her home gateway and the ID of an alternative node (or *mirror*) in the unlikely case that the home gateway is not reachable. Note that, unlike several previous works (e.g., [13], [14], [18]), this mirror is selectable by the user herself.

Also note that in contrast to some state-of-the-art solutions (e.g., [13]), a user only publishes pointers to her data, while the data itself is kept under physical control of the user. Directly storing data in the DHT would have several undesirable consequences [3], the most important of which would be that users would not have any control over the location(s) at which her data would be stored.

2) SocialGate Applications: SocialGate is running as a middleware on home gateways. At the same time, social applications typically reside on user devices. SocialGate is not limited to run a single application (e.g., a microblogging application), but can in fact grant access to user data to arbitrary applications. For instance, we have implemented a web-based application similar to Facebook that communicates with the middleware via a simple web interface. However, developers can also create desktop applications exploiting the middleware.

3) SocialGate Data Privacy: SocialGate employs Attribute Based Encryption (ABE) [16] to facilitate access control to user data. ABE has been widely used in DOSNs for the same purpose and ensures that users can define fine-grained access control to their data based on attributes assigned to other DOSN users [3]. The main idea is that only users holding the appropriate attributes (e.g., 'family member' or 'co-worker')—encoded in cryptographic keys—can access the subset of a user's data that is assigned to combinations of these attributes. For instance, a user can setup a policy that allows keys containing the 'family member' attribute to access holiday pictures, while not allowing this for keys that hold the 'co-worker' attribute.

4) SocialGate Mirror Selection: As mentioned previously, home gateways are typically highly available devices. In a preliminary step to evaluate the feasibility of SocialGate we collected one month of real-world uptime data of 456 routers from the Freifunk network in Goettingen, Germany[3].

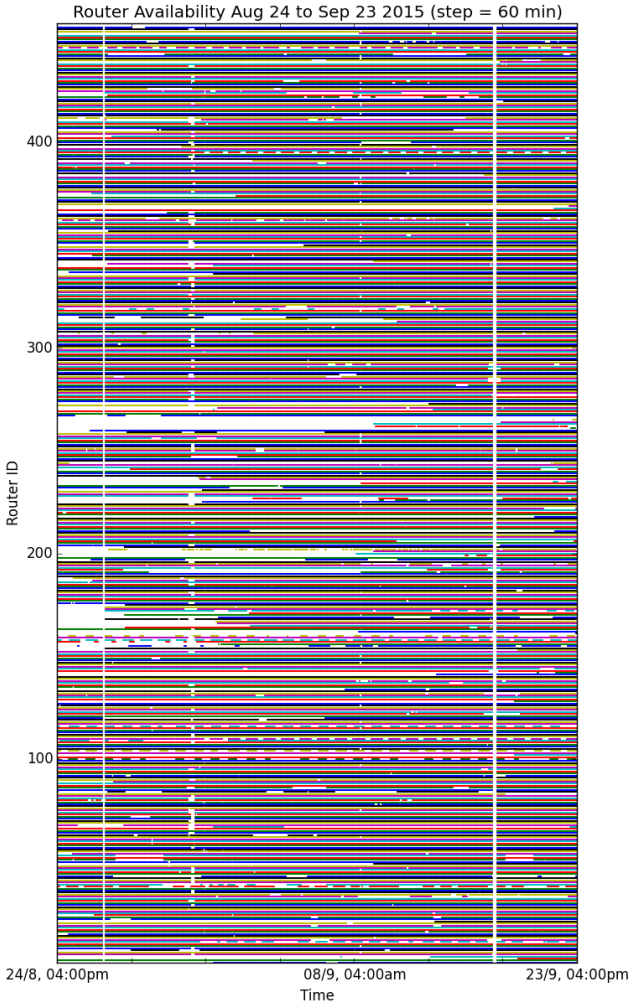

Fig. 2. Router availability in the Freifunk network of Goettingen, Germany. The three major outages are measurement errors.

Figure 2 shows the hourly uptime measurement for these devices for the entire measurement period, where white areas represent offline time for a particular router ID. We observe that while routers are generally highly available, they also experience offline periods. Some users even turn of their routers over night—Figure 3 shows this behaviour for a subset of routers.

These observations necessitate the mirroring of social data. Otherwise, data requests towards a user u's router during his offline periods would be unsuccessful. With a mirror in place, requests to u can be handled by that mirror in the absence of

[3]http://www.freifunk-goettingen.de

Fig. 3. Several users turn off their devices during night time, leading to less availability of these devices. Each line in the above plot shows the periods, in which a device was measured online (coloured) or offline (blanks) for the entire measurement period. Note that the plot shows a random subset of devices for which we observed diurnal patterns.

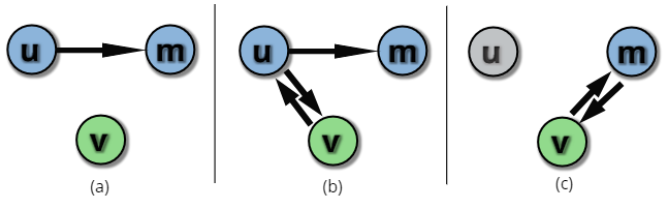

Fig. 4. Interaction with mirrors in SocialGate. (a) Initially, u selects a mirror m. (b) Then, once u updates her data locally (e.g., after receiving a message from v), u also updates her data at m. (c) Now, if u goes offline, v can retrieve u's data from m and send updates to that node as well (c).

u's router (see Figure 4 for an illustration). This is the case for both simple requests for data (e.g., retrieval of the latest status of u), but also for writing data towards u (e.g., sending a message to u). In the latter case, the mirror will store the encrypted message as an update to u's data, u will collect the update after returning online, and apply the update (decrypt the message) locally. Note that in general, all data of u stored at the mirror is stored in encrypted form. Thus, other users, including the mirror itself, still need to be eligible (by their ABE attributes) to access the data.

Since devices are available for 91.5% of the time on average, by default SocialGate will select only one mirror per user (we will show in our evaluation that this is indeed sufficient). The information about that mirror is stored in the DHT overlay.

While this mirror is ultimately selected by the user, SocialGate does provide an initial recommendation of high-quality mirrors. This recommendation employs logistic regression to predict, based on previously measured reachability of other devices, whether or not a mirror candidate will be online during a user's offline periods. This information is obtained by each user by traversing other nodes in the network (e.g., within DHT requests). Applied to our collected measurement dataset, we obtain a mean squared error of 0.0020 for these predictions. The approach works particularly well in identifying diurnal patterns of users.

B. Challenges

Implementing an infrastructure substrate for large-scale social data management on home gateways has a wide range of benefits as discussed above. However, it also induces serious challenges that need to be addressed in the system design.

1) Router Access: First and foremost, we cannot assume each user to have access to a home router. We thus designed SocialGate to also run on standard operating systems, which allows usage by users who do not want to contribute their home routers or cannot access one. In these cases, users can run SocialGate on desktop PCs, laptops or even rented cloud instances (e.g., Amazon EC2), if desired. This generic implementation also facilitates the contribution of server nodes by users, such as seen in Diaspora[4] or Mastodon, to further improve overall system availability.

2) Storage Space: We conducted a survey on six large service providers in both Germany and the US and found storage capacities between 4MB and 512MB for the standard routers handed out by these ISPs, which is clearly insufficient for storing OSN data. Fortunately, now all but one ISPs hand out routers equipped with USB ports, allowing for cheap (about US$ 1 per GB) storage space. Note that, contrary to cloud subscription models, this is a small one-time investment.

3) Ease of Usability: As not all provider routers allow third-party software to be installed on their pre-configured OS, we have built SocialGate based on OpenWrt.[5] An automated script allows technically un-savvy users to install SocialRoute on their routers. Furthermore, our SocialGate demo application does not require users to start a client on their devices of daily use (e.g., smartphones), but rather uses a web interface for communicating with the underlying overlay.

IV. Implementation

We have implemented SocialGate in Python for OpenWrt, and have made the source code available for download.[6] SocialGate can be run on any of the more than 600 devices currently supported by OpenWrt. Additionally, as discussed previously, SocialGate works on any operating system supporting Python as well.

For the DHT overlay, we chose Entangled[7], a Python implementation of Kademlia [22]. We have further implemented a multitude of typical OSN features in a demo application for SocialGate, including content sharing and commenting as well as instant messaging. Figure 5 shows a basic SocialGate profile in the Django-based web interface of the demo application

[4]http://www.joindiaspora.org
[5]http://www.openwrt.org
[6]https://github.com/DimaGitter/SocialRoute
[7]http://entangled.sourceforge.net

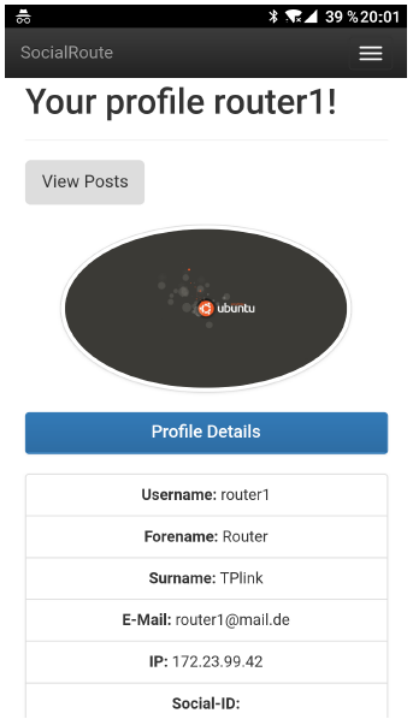

Fig. 5. SocialGate demo application web interface. This screen shows a basic user profile.

running on a smart phone, which does not require the user to install any software on her phone.

V. Evaluation: Simulation

As discussed above, the two most important metrics for evaluating DOSNs without server involvement are *data availability* (i.e., the percentage of *retrievable* user profiles in the network at a specific time t) and *mirroring overhead* (i.e., the number of mirrors required to achieve the data availability at time t). Usually, optimizing one of these metrics is only possible at the cost of the other: increasing the amount of mirrors typically makes data more available throughout the system, but also increases the overhead to, for instance, synchronize the replicas stored at the set of mirrors [3].

Figure 6 shows the performance of SocialGate with regards to these two metrics. Here, we run different simulations of SocialGate with the real-world uptime data from Freifunk as a basis. Mirrors are recommended based on our logistic regression predictor. We limit the amount of replicas stored at a single node to five in order to prevent highly-rated mirrors to become overloaded.

We distinguish three different simulations, and run each of them five times. In the first experiment, we do not allow any mirroring, but only check whether each router is online at all times (*no mirror* in Figure 6). We see that SocialGate can achieve a data availability of $\approx 95\%$, which is already higher than many cooperation-based systems [11], [12]. The significant drops at $\approx$ 200 and 400 hours are likely due to

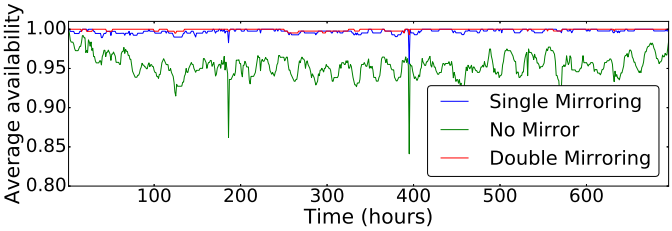

Fig. 6. SocialGate Data Availability. With a single mirror, SocialGate is able to achieve 99.7% average data availability.

measurement errors in the original trace. Also, the availability is oscillating around the 95% marker regularly on a daily basis. The reason for this behavior is that, although routers are stable devices, they still experience online and offline patterns. In particular, we can see the day and night patterns as shown above—more devices are available during the daytime.

If we introduce one mirror per user (*single mirroring* in Figure 6), the data availability increases to 99.7% on average. SocialGate in this case beats comparable P2P approaches by far, as those typically employ 4 to 13 replicas, while still falling short of this benchmark availability [11], [12], [14]. Note that in terms of overhead our result is close or in some cases even equivalent to that in systems using cloud services for enabling data availability—in these systems, users often mirror their data exactly once as well, i.e., on the cloud service [7].

Additionally, we can improve data availability even further by adding one additional mirror(*double mirroring* in Figure 6). In this case, we can push data availability to 99.97% (or more than "three nines" availability). However, this comes at additional expenses: If we introduce a second mirror, we need to take care of synchronization among these mirrors. We leave the design of appropriate procedures for future work.

Both single mirroring and double mirroring availability rates are therefore higher than, for instance, at least 13% and 47% of the top 1000 most visited websites in the US, respectively.[8] Note that while our availability rates are not calculated in stress tests that may reduce data availability by pushing nodes to their boundaries, they are still indicators of SocialGate's superior performance when compared to related work.

VI. Evaluation: Deployment

As a final step, we have deployed our reference implementation in a small testbed consisting of three TP-Link TL-MR3020 routers (enabled with OpenWRT) and four user devices (laptops, desktop PCs and smartphones). Figure 7 shows an overview of the deployment. Our goal was not to investigate the large-scale data availability or mirroring overhead, but rather to analyze aspects that directly affect user experience as well. Next to data availability itself, user experience in a distributed system is heavily influenced by the

[8]According to an uptime measurement by CloudEndure: https://www.cloudendure.com/blog/top-100k-websites-downtime-report-2016-q1/

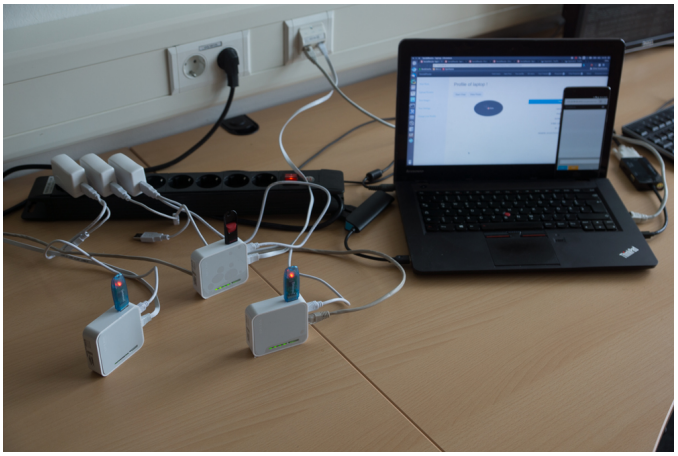

Fig. 7. Local testbed for SocialGate. The three routers and the user devices are interconnected in a local wireless network.

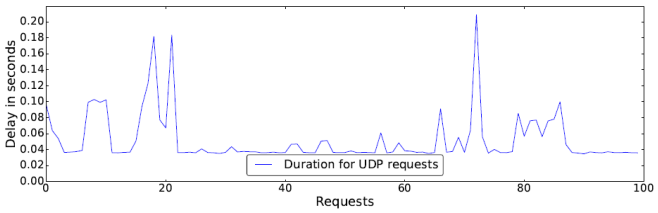

Fig. 8. Delay of UDP requests in SocialGate

latency experienced by the users. Even though data availability may be high, requesting data from distributed, embedded devices may be slow.

In SocialGate (and the underlying Kademlia DHT) all requests (e.g., requesting user IDs or IP addresses) other than the direct file transfer are UDP requests. Figure 8 shows an exemplary snapshot of delay for a sequence of 100 different UDP requests. We observe that most requests take about 40ms to be processed, while peaks can rise to 200ms in the worst case. Note that these measurements, due to the nature of our local deployment, do not include significant delay between nodes. However, even taking increased RTTs for UDP requests (e.g., for transatlantic requests) into account, the delay for a request should be well within a few hundred milliseconds.

VII. Conclusion and Future Work

In this work we have introduced *SocialGate*, an approach to build an infrastructure surrogate for decentralized online social networks based on home gateways. SocialGate offers a series of benefits over previous approaches and our evaluation based on real-world data shows that it performs well. In particular, it can achieve an average data availability of 99.7% by using a single replica of user data, keeping the replication overhead low. In future work, we will focus on advanced mirroring strategies, evaluate SocialRoute in more detail (e.g., when only fractions of users employ home routers), provide replica synchronization methods to facilitate double mirroring, and deploy our prototype on a large scale.

References

[1] M. Zimmer, "Mark Zuckerberg's theory of privacy," http://wapo.st/1gJOqEu (all links have been checked on August 1st 2017), February 2014.

[2] E. Barnett, "Facebook settles lawsuit with angry users," bit.ly/YYo1eZ, May 2012.

[3] D. Koll, J. Li, and X. Fu, "The Good Left Undone: Advances and Challenges in Decentralizing Online Social Networks," *Elsevier Computer Communications*, April 2017.

[4] S. Buchegger and A. Datta, "A Case for P2P Infrastructure for Social Networks - Opportunities and Challenges," in *Proceedings of the 6th International Conference on Wireless On-Demand Network Systems and Services (WONS 2009)*. IEEE, 2009, pp. 161–168.

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

[6] "The Diaspora Project," https://joindiaspora.com/, 2010.

[7] S.-W. Seong, J. Seo, M. Nasielski, D. Sengupta, S. Hangal, S. K. Teh, R. Chu, B. Dodson, and M. S. Lam, "Prpl: A decentralized social networking infrastructure," in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, ser. MCS '10. New York, NY, USA: ACM, 2010, pp. 8:1–8:8.

[8] A. Shakimov, H. Lim, R. Cáceres, L. P. Cox, K. Li, D. Liu, and A. Varshavsky, "Vis-á-Vis: Privacy-preserving Online Social Networking via Virtual Individual Servers," in *Proceedings of COMSNETS 2011*, 2011.

[9] P. Stuedi, I. Mohomed, M. Balakrishnan, Z. M. Mao, V. Ramasubramanian, D. Terry, and T. Wobber, "Contrail: Decentralized and privacy-preserving social networks on smartphones," *IEEE Internet Computing*, vol. 18, no. 5, pp. 44–51, 2014.

[10] E. Klukovich, E. Erdin, and M. H. Gunes, "Posn: A privacy preserving decentralized social network app for mobile devices," in *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*. IEEE, 2016, pp. 1426–1429.

[11] K. Rzadca, A. Datta, and S. Buchegger, "Replica Placement in P2P Storage: Complexity and Game Theoretic Analyses," in *Proceedings of the 30th IEEE International Conference on Distributed Computing Systems (ICDCS 2010)*. IEEE, 2010, pp. 599–609.

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

[13] S. Nilizadeh, S. Jahid, P. Mittal, N. Borisov, and A. Kapadia, "Cachet: A Decentralized Architecture for Privacy Preserving Social Networking with Caching," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies (CoNEXT 2012)*. ACM, 2012, pp. 337–348.

[14] D. Koll, J. Li, and X. Fu, "Soup: An online social network by the people, for the people," in *ACM/USENIX Middleware 2014*.

[15] A. De Salve, B. Guidi, P. Mori, L. Ricci, and V. Ambriola, *Privacy and Temporal Aware Allocation of Data in Decentralized Online Social Networks*. Springer International Publishing, 2017, pp. 237–251.

[16] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in *Proceedings of the 28th IEEE Symposium on Security and Privacy (S&P 2007)*. IEEE, 2007.

[17] D. Liu, A. Shakimov, R. Caceres, A. Varshavsky, and L. P. Cox, "Confidant: Protecting OSN Data without Locking it Up," in *Proceedings of the 12th ACM/IFIP/USENIX International Middleware Conference (Middleware 2011)*. Springer, 2011, pp. 61–80.

[18] R. Sharma and A. Datta, "SuperNova: Super-peers Based Architecture for Decentralized Online Social Networks," in *Proceedings of the 4th IEEE International Conference on Communication Systems and Networks (COMSNETS 2012)*. IEEE, 2012, pp. 1–10.

[19] T. Paul, D. Puscher, and T. Strufe, "The user behavior in facebook and its development from 2009 until 2014," *arXiv preprint arXiv:1505.04943*, 2015.

[20] M. Marcon, B. Viswanath, M. Cha, and K. P. Gummadi, "Sharing Social Content from Home: A Measurement-driven Feasibility Study," in *NOSSDAV 2011*.

[21] R. Mortier, J. Zhao, J. Crowcroft, L. Wang, Q. Li, H. Haddadi, Y. Amar, A. Crabtree, J. Colley, T. Lodge *et al.*, "Personal data management with the databox: What's inside the box?" in *ACM CAN Workshop*, 2016.

[22] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *IPTPS 2001*.