On the Privacy and Performance of Mobile Anonymous Microblogging

Marius Senftleben, Ana Barroso, Mihai Bucicoiu, Matthias Hollick, *Member, IEEE*, Stefan Katzenbeisser, *Senior Member, IEEE*, and Erik Tews

***Abstract*— Microblogging is a popular form of online social networking activity. It allows users to send messages in a one-to-many publish-subscribe manner. Most current service providers are centralized and deploy a client–server model with unencrypted message content. As a consequence, all user behavior can, by default, be monitored, and censoring based on message content can easily be enforced on the server side. A distributed, peer-to-peer microblogging system consisting of mobile smartphone-equipped users that exchange group encrypted messages in an anonymous and censorship-resistant manner can alleviate privacy and censorship issues. We experimentally evaluate message spread of such systems with simulations that run on a range of synthetic and real-world mobility inputs, thus extending the previous work. We show that such systems are feasible for a range of mobility and network settings, both under normal and under adversarial conditions, e.g., under the presence of nodes which jam the network or send spam.**

***Index Terms*— Microblogging, anonymity, censorship-resistance, peer-2-peer, mobile networking, simulation.**

I. Introduction

MICROBLOGGING is a popular form of Online Social Networking (OSN) activity. It is part of an area of OSN known as micromedia, where users share small snippets of content media such as text, images or video. Specifically, we focus on microblogging of small text messages. Users can subscribe to another user's channel or create message channels themselves. A service broadly used to this end is *Twitter*.

Client-server architectures back most of the existing microblogging services, and hence message content is visible to service operators. Confidentiality is thus not addressed and service providers might analyze all user-generated content. Moreover, not only message content but the whole user activity is transparent to providers, for instance, received and sent messages as well as group subscriptions.

Another issue is the susceptibility of the servers for censorship. All messages are relayed over them and messages can thus be stricken out depending on their content, sender, or receiver, hence interfering content spread according to some censorship policies. Furthermore, centralized systems are vulnerable to local Internet outages, either unintentionally, e.g., power grid failures, or intentionally, e.g., Internet shut-downs via 'kill-switches'. In both cases such systems cease to function.

To overcome the aforementioned issues, and based on the proposals presented in [1]–[3], we thoroughly evaluate and analyze the feasibility of a mobile distributed microblogging system consisting of users carrying their mobile devices. We extend the work of [1], whose system goals are anonymity, message confidentiality, and censorship-resistance. All messages are encrypted under a group key, and they are stored across all nodes in a decentralized fashion. In particular, we evaluate a system where message propagation is carried out using local, point-to-point communications taking place between close pairs of nodes; stressing that this is a middleware-free scenario without orchestrating entities.

Our contributions are as follows:

- We experimentally show the feasibility of our mobile microblogging system using simulations with both synthetic and empirical mobility datasets.
- We simulate different networking setups to gain insights on how they affect message spread.
- We evaluate the system under adversarial conditions, discuss its privacy, and state lessons learned.

The rest of the article is structured as follows: an overview of the system is given in Section II. Section III details the different synthetic and empirical mobility datasets used to conduct the simulations, followed by the evaluation of the simulation results in Section IV. Section V deals with the system's performance under adversarial conditions. In Section VI we discuss our findings. Section VII treats related work, and Section VIII concludes the paper.

II. System Overview

We consider microblogging systems implemented as a decentralized, distributed peer-to-peer network made up of mobile nodes, similar to those described in [1]. Nodes consist of humans carrying a mobile device (such as a smartphone), who create microblogging messages and propagate them using

Manuscript received September 21, 2015; revised December 31, 2015 and February 12, 2016; accepted February 14, 2016. Date of publication March 14, 2016; date of current version April 12, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jianying Zhou. *(Marius Senftleben and Ana Barroso contributed equally to this work.)*

M. Senftleben, A. Barroso, M. Hollick, S. Katzenbeisser, and E. Tews are with the Center of Advanced Security Research Darmstadt, Department of Computer Science, Technical University of Darmstadt, Darmstadt 64289, Germany (e-mail: senftleben@seceng.informatik.tu-darmstadt.de; abarroso@seemoo.tu-darmstadt.de; matthias.hollick@seemoo.tu-darmstadt.de; katzenbeisser@seceng.informatik.tu-darmstadt.de; e_tews@seceng.informatik.tu-darmstadt.de).

M. Bucicoiu is with the Department of Computer Science, Politehnica University of Bucharest, Bucharest 060042, Romania (e-mail: mihai.bucicoiu@cs.pub.ro).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2016.2541633

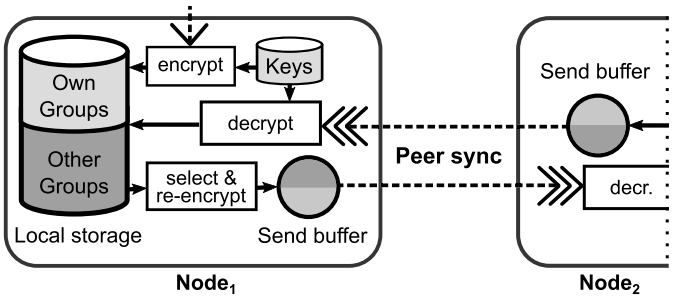


Fig. 1. Schematic overview of the microblogging system.

peer syncs, i.e., direct wireless links established between pairs of nodes.

We use the schematic overview depicted in Figure 1 to describe the functioning of the system: nodes belong to groups, and each node has an asymmetric group key for each of the groups it belongs to, in order to be able to encrypt and decrypt messages belonging to those groups. These keys are stored in a node's key store, e.g., the *keys* of Node_1.

Group keys are exchanged beforehand based on existing social relationships and trust. When nodes physically encounter, they can extend group memberships passing groups keys to one another, so that newcomers can join already existing groups. Therefore, key management and key exchange need to be addressed. The key exchange can, for example, be conducted by using near field communication (NFC), or by using a quick response code (QR-Code). This facilitates joining a group reliably and reduces misuse of group keys. However, there are still some inevitable risks since our system has been designed to be an open one.

Each node maintains its *local storage* of encrypted messages, containing messages not only of its *own groups* but also of those *other groups* it is not affiliated to, and thus unable to decrypt. All messages are encrypted under a public group key. In this way, a decentralized storage of messages is created across all nodes. Any node can *encrypt* a newly created message by using one of its public group keys, and put it in its local storage for further propagation. New messages created by a node are also directly put in the send buffer once they are created to effectuate a timely dissemination of the new message.

Send buffers contain the messages to be exchanged in peer syncs. These messages are already encrypted, and are re-encrypted each time before being placed in the send buffer to guarantee message unlinkability. Prior to each peer sync a send buffer is filled following a fixed synchronization strategy, which determines how nodes select the messages that are to be placed in the send buffer out of their local storages. We deploy two different sync strategies: *Random (RD)* and *Prioritized (PR_p)* peer-syncing. For RD, all slots in the send buffer are filled uniformly at random out of all messages in the local storage. For PR_p , a fraction $0 \leq p \leq 1$ determines the amount of slots reserved for messages from groups the node is a member of. $\text{PR}_{0.4}$ syncing thus means that 40% of the send buffer is filled with messages from groups the node is a member of, and 60% is filled uniformly at random with messages from other groups. This PR_p synchronization strategy allows to prioritize own messages.

Peer syncs propagate the messages and are initiated whenever two nodes are in proximity of each other. During a peer sync two nodes, i.e., Node_1 and Node_2, bidirectionally exchange their prepared send buffers. This happens through a direct wireless point-to-point link established between the nodes for this purpose once they are in communication range. Upon receiving a send buffer, a node has to *decrypt* each message and check if it belongs to one of the groups it is a member of by using its private group keys. Then the messages are sorted into its local storage. Thus, messages are eventually distributed across the nodes within the network. Since in this delay-tolerant networking setting direct node encounters are, by design, the only means of propagating messages, the movement patterns of the nodes are responsible for how fast and to what extent messages spread. Sparsely connected movement patterns will only see a message spread as reflected by the number of node encounters that take place. Therefore, if the number of node encounters is lower, then the resulting message spread will also be lower.

To achieve unlinkability—and thus anonymity—all nodes have to *re-encrypt* the encrypted messages each time before being resent in a peer sync. The cryptographic scheme must allow any node to re-encrypt any already encrypted message, regardless of whether the node is in possession of that specific group key pair or not. Any node is thus able to re-encrypt any message—from both its group messages as well as others—only by using the ciphertext of the encrypted message, and is able to transform it into a new ciphertext; a process that can be iterated repeatedly. Re-encryption prevents the tracing of messages by making them unlinkable, i.e., the ciphertext of a message received by a node is always different from the ciphertext of the same message being sent out by a node in a peer sync. To decrypt a message a node has to check to which group the message belongs to by brute forcing with all the group keys it has, since the ciphertexts do not contain group IDs to guarantee anonymity. However, for each key only a singular decryption is needed to obtain again the plaintext of a re-encrypted message, regardless of how many times it has been re-encrypted before. To that end, a cryptographic scheme capable of universal re-encryption such as [4] has to be used.

Sender-receiver anonymity is thus achieved unlinking in- and outgoing messages relayed from node to node via peer-syncing, and selecting messages pooled in the local storages. On the other hand, confidentiality is accomplished encrypting all messages. Finally, censorship-resistance relies on the decentralized, distributed nature of the peer-to-peer system.

III. Mobility Evaluation

To measure the performance of the microblogging system, we conduct discrete, event-based simulations that emulate the behavior of smartphone users. Since nodes only exchange a message when they physically encounter other nodes, the mobility input is a crucial ingredient: the contacts between nodes determine the underlay network on top of which the communication takes place, and have an impact on the success of spreading a message across the network.

In this section we analyze the mobility input to our simulation. We use opportunistic networking metrics to characterize

the input mobility scenarios, so that we can describe the conditions under which our system is functional in terms of the achieved message spread, the core metric we use to characterize the proposed scheme.

We use two types of mobility data: real-world data collected in three different scenarios, and synthetic data generated using the BonnMotion [5] mobility framework. With the first we try to understand the behavior of our communication scheme under various real-world conditions. Since the collected real-world data does not represent the whole variety of application scenarios, we also rely on synthetic mobility data in order to determine the *networking conditions*, such as network density and average number of neighbors, within which the architecture is functional.

A. Opportunistic Networking Metrics

The performance of the message distribution scheme is heavily dependent on the opportunities that nodes have to meet and exchange data. In order to understand the impact of mobility on the performance of our communication system, we analyzed the datasets according to the following metrics:

- Node degree is the number of neighboring nodes that a node has contact with, counted in periods of two hours.
- Contact duration indicates how long a pair of nodes meet (in second)s.
- Inter-contact duration indicates how long a node is not in contact with any other node (in seconds).
- Contacts per node per day is the number of meetings per day a node participates in.

B. Empirical Mobility Data

To evaluate the system we used three real-world mobility datasets:

- The Bluetooth dataset [6] contains the time intervals in which 285 smartphone users are in range of each other using Bluetooth as a communication technology. The dataset spans a period of 120 days and was mostly collected during an event at a congress center.
- The Nokia Mobile Data Challenge (NMDC) [7] contains the GPS traces of 186 smartphone users over a period of 576 days. The total gathered data spans over an area of 224km times 344km. Most of the nodes remained in the Lausanne region but some traveled as far as the center of France, hence the large area.
- The SUVnet Traces [8] contains 30 days of mobility traces of 4445 taxis from Shanghai and surrounding areas. The GPS traces span over an area of approximately 193km times 222km. The data was obtained from the Wireless and Sensor Networks Lab of Shanghai Jiao Tong University.

Both the NMDC and SUVnet are based on the GPS traces of mobile nodes. These traces were processed following the BonnMotion [5] mobility framework in order to obtain a list of meeting intervals for each pair of nodes, from which we calculate the aforementioned metrics. Based on the timeline of the recorded GPS traces, BonnMotion interpolates the path of the nodes and infers when and for how long any two nodes meet. The Bluetooth dataset already contained the meeting intervals measured on the nodes.

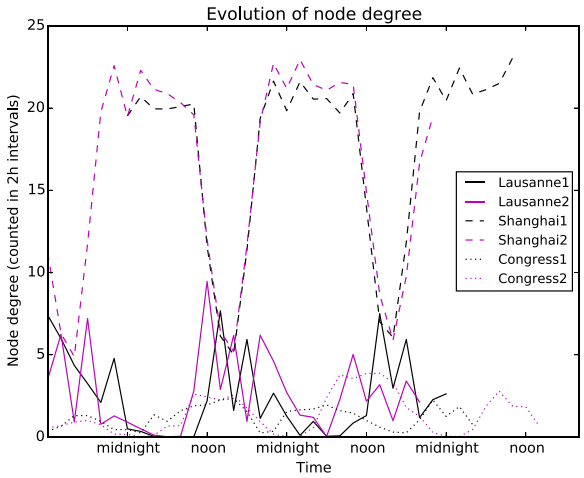


Fig. 2. Node degree, counted in 2h slots, changing over time due to node mobility.

TABLE I
AVERAGE NODE DEGREE AT DIFFERENT TIMES OF THE DAY

	Daytime	Night-time	Outlier groups
Congress	approx. 3 to 4	approx. 0 to 2	up to 25
Lausanne	approx. 4 to 7	approx. 0 to 2	up to 70
Shanghai	approx. 5 to 10	approx. 20 to 24	up to 200

From each dataset we extracted two sample sets in different time frames. Our criterion was to select a range of a few days with a high number of active nodes, since the participation rate in the data collection process fluctuated. The entire datasets run for several months, which is unnecessarily long for our analysis. The used data samples are as follows:

- Congress1 and Congress2, extracted from the Bluetooth dataset: approximately 3.25 days each, 40 and 162 nodes respectively
- Lausanne1 and Lausanne2, extracted from the NMDC dataset: 2.5 days each, 141 and 137 nodes respectively.
- Shanghai1 and Shanghai2, extracted from the SUVnet Traces: 2.5 days each, 4366 and 4347 nodes respectively

For our analysis, we assume two nodes are in communication range and can be considered neighbors when the distance between them is smaller than 15m. For practical purposes, this distance is technology-dependent.

In the following paragraphs we describe the results of our mobility analysis, which allow us to understand how node movement affects message dissemination. A summary of relevant statistics can be found in Table II for comparison.

1) Node Degree: To measure node degree we counted the number of neighbors of each node over 2-hour periods. All datasets exhibit a 24h periodic pattern, as can be seen in Figure 2. We summarize the relevant results in Table I.

During daytime, node degree varies greatly in the Shanghai and Lausanne datasets. In Shanghai, some nodes reached a 2h-count of 200 neighbors, possibly due to parking taxis over night. In Lausanne there was a consistent group of outliers with 60 to 70 neighbors in some peak times, which

TABLE II
CONNECTIVITY STATISTICS OF THE DATASET SAMPLES

	Congress1	**Congress1**	**Lausanne1**	**Lausanne2**	**Shanghai1**	**Shanghai2**
Node count	40	162	141	137	4366	4347
Meeting count	1332	7057	5826	5223	1585597	1412046
Total duration [s]	280885	281114	216000.0	216000.0	216000.0	216000.0
Node degree (ND) mean	1.19	1.39	2.49	2.73	17.05	15.56
ND standard deviation	1.82	3.17	8.47	7.68	18.43	17.46
ND median	0	0	0	0	11	9
Meeting duration (MD) mean [s]	420.1	3846.0	7406.0	6885.6	15.5	17.0
MD standard deviation [s]	1255.9	853.0	30495.9	21201.0	199.9	330.3
MD median [s]	150.0	150.0	234.1	254.3	2.9	2.8
Average meetings per node	33.3	43.6	41.3	26.6	363.2	324.8
Average meetings per day	409.7	2169.0	2330.4	2089.2	63423.8	564818.4
Average meetings per node per day	10.2	13.4	16.5	15.2	145.3	129.9

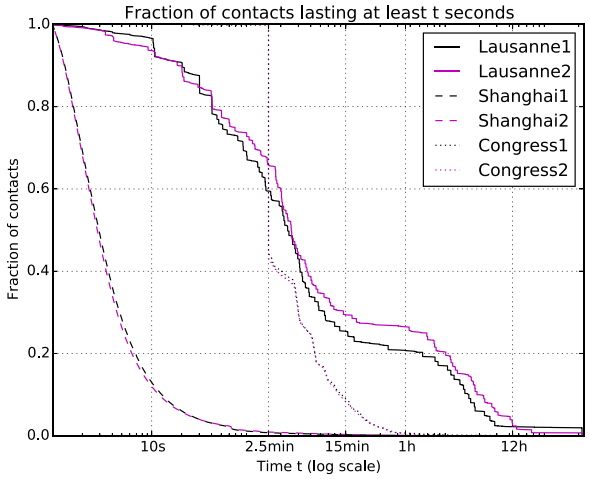


Fig. 3. Distribution of the contacts lasting longer than t time. Observe that in the Shanghai dataset only 10% of the contacts take more than 10 seconds, all Congress contacts take between 2.5 and 60 minutes and, in Lausanne, approximately 60% of the contact lengths vary greatly below the 15 minute mark, while 20-25% take longer than 1 hour.

we attribute to university lectures. The network graphs of Lausanne, Shanghai and Congress2 showed that some groups of nodes tended to meet often but many others had few neighbors, which explains the high node degree dispersity shown in Table II. In Congress1, the network graph was more homogeneous, resulting in a low deviation from the mean.

In the Congress dataset the node degree does not vary significantly over the day, possibly because there were events taking place until late night and attendees could be in the building any time of the day or night. In Congress1 some nodes reached 6 to 8 neighbors in a 2h-interval and in Congress2 some reached as many as 25 neighbors, potentially during a popular event.

2) Contact Duration: As depicted in Figure 3, the Congress dataset only contains contacts longer than 2 minutes. Most last less than 7 minutes, with some reaching 15 minutes.

In the Lausanne datasets, approximately 40% of the contacts are shorter than 2 minutes. We also observed that a significant number of encounters (20%–30%) lasts longer than 1h. Given that some participants were university students, this might correspond to lecture periods.

Almost 90% of the contacts in the Shanghai scenario are shorter than 10s, probably due to the vehicular nature of the network, where nodes tend to cross each other's paths at high speed. Most of the remaining 10% of the meetings are shorter than 2 minutes, possibly representing traffic jams or stops at taxi centrals. Only a small fraction of the meetings last a longer period.

3) Inter-Contact Duration: In all datasets most nodes can be split into two groups: those that are almost always in contact with at least another node, and those that are almost always isolated. For comparison, we list the percentage of nodes with typical inter-contact times of less than 15 minutes (frequent contact) against that of nodes isolated longer than 24 hours on average (strong isolation):

- 50% with frequent contact vs. 20-30% with strong isolation in the Congress dataset,
- 40-50% with frequent contact vs. 30-40% with strong isolation in the Lausanne dataset,
- and 60-90% with frequent contact vs. 10-20% with strong isolation in the Shanghai dataset.

Higher percentages of isolated nodes for a long period of time, like in the Lausanne dataset, impact negatively the ability of the nodes to spread new messages quickly.

4) Contacts per Node per Day: The number of contacts per day strongly influences a message's likelihood of reaching new nodes. The three groups of dataset samples exhibit very different behaviors:

- In the Congress dataset, the vast majority of the nodes has less than 100 contacts a day. In Congress1 the distribution of the number of daily contacts is quite homogeneous between 0 and 100, with a slight increase between 40 and 60 contact per node. In contrast, in Congress2 there is an inverse relation between the number of nodes and how many daily contacts they have, ranging from 12% of the nodes meeting less than 20 times a day to 2% of the nodes meeting from 80 to 100 times a day. On average, a node from Congress1 has 10 daily contacts, and in Congress2 that number rises up to 13.
- In both Lausanne datasets there is a significant number of nodes (approximately 30%) meeting less than 10 times a day. No nodes have more than 100 daily contacts.

On average, each day a node has approximately 17 contacts.

- The nodes from the Shanghai dataset meet much more often than the ones from the previous datasets. While there are quite a few nodes (about 8%) with less than 100 contacts per day, there is a significant number of nodes (approximately 20%) that meet 500 to 2000 times a day. On average, a node from Shanghai1 has 145 daily contacts, while a node from Shanghai2 has 130.

Considering only both inter-contact duration and the average number of contacts per day of a node, the Shanghai dataset may provide the best communication opportunities. However, the very short contact duration of the Shanghai dataset is detrimental to the success of the communication scheme with technologies that require a setup phase. In the remaining scenarios, Lausanne exhibits longer meeting durations and a higher node degree than Congress, which we expect to give an advantage in message dissemination, however Congress has a lower percentage of isolated nodes that may compensate the difference.

C. Synthetic Mobility Data

Even though not every inhabitant of a city with a mobile device will participate in mobile microblogging, the node densities of the sampled data are orders of magnitude below the densities of world cities. Additionally, the real world datasets we used target specific subcommunities, and are therefore not representative of all the possible usage scenarios that our system may encounter. We must therefore evaluate mobile microblogging over a wider range of conditions using synthetic mobility models, in order to detect critical points at which the scheme may stop working.

For the evaluation we generated mobility data with the *ManhattanGrid* [9] and the *GaussMarkov* [10] models provided by the BonnMotion framework [5]. We varied the number of nodes from 25 to 400 in steps of 25. The areas ranged from 1 to 70 km^2 in steps of 0.5km^2. We consider it is a reasonable range of synthetic data for this application, since we only expect a subset of the population to use mobile microblogging. We are therefore not attempting to emulate real cities, which can be much more dense, but to determine a density threshold from which point onwards our system is feasible.

In both models, node degree decreases sharply for areas higher than 10 km^2. While an increased number of nodes always raises the node degree, this is more pronounced from 1 to 5 km^2. *GaussMarkov*'s average node degree varies between 0.0002 and 0.28, while *ManhattanGrid*'s ranges from 0.0002 to 2.

The number of contacts per node per day strongly follows the pattern of the node degree. For the *GaussMarkov* model, this value ranges between 2.7 in the least dense scenario and 59278.2 in the most dense. In *ManhattanGrid*, it varies between 0.6 and 71321.1 contacts per node per day.

Varying the area and the number of nodes shows little effect on contact durations. For *GaussMarkov* they range from 17.5s to 22.5s on average, and from 17.6s to 29.5s for *ManhattanGrid*.

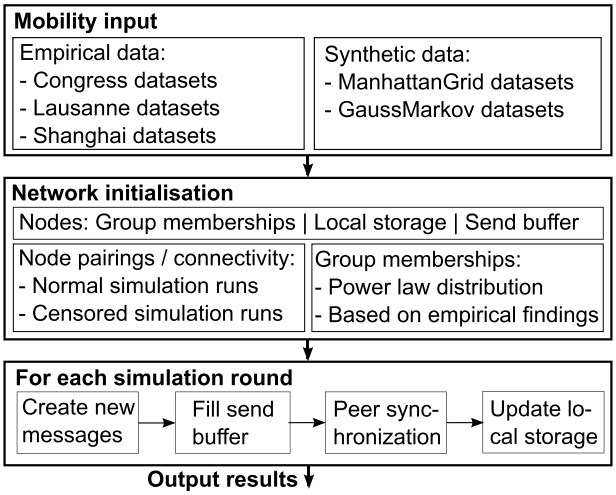


Fig. 4. Overview of the simulation.

IV. Simulation of the Microblogging System

The mobility input determines the network on which the simulated microblogging system runs. The goal of the simulation is to gain insights on message propagation using real empirical data as well as synthetic data. Parameterized link settings are introduced to identify bottlenecks. We first characterize the simulation runs and then present the results thereof.

A. Simulation Setup

The message propagation is assessed with a discrete, event-based simulation, which is run in rounds of 5 minutes duration, the time in which two Bluetooth pairings can be conducted. An overview of the simulation is given in Figure 4.

Taking the empirical and synthetic mobility datasets analyzed in the preceding section as input, we simulate the group assignments, message creation events and peer synchronizations of the nodes.

1) Group Assignment and Message Creation Events: Real microblogging user behavior data serves as a basis for group formation and message creation events [11]. The overall pattern of group message creation events follows power law distributions [12]. During simulations the group memberships are static, i.e., there is no node churn.

2) Peer Syncs and Local Storage: The mobility patterns are the foundation used to derive node connectivity. We determine whether a given pair of nodes is eligible for a peer sync by setting the maximum distance over which a peer sync can take place to 15 meters based on Bluetooth characteristics. A peer sync might also occur in those cases where contact duration is too short for current technologies to establish a link, e.g., the vehicular movements in the Shanghai datasets (cf. Fig. 3). This is not the case for the Congress datasets, in which all links established are based on actual Bluetooth pairings with successful data transfers that took place during data collection. Each peer sync makes a pair of nodes exchange a send buffer with a specified number of messages drawn according to a fixed synchronization strategy. Each node's local storage saves up to 10,000 messages and the latest

incoming messages from a peer sync shift out the oldest messages received (cf. Section II).

3) Parameterized Link Settings: Λ_p^b is introduced as shorthand notation for parameterized link settings, with p being the maximum allowed number of peer syncs per round and node, and b being the size of the send buffer exchanged per peer sync. Various link settings are used to assess their effects on message spread and to identify bottlenecks. *Bluetooth* ($\Lambda_2^{0.1k}$) link settings limit the number of peer syncs to 2, i.e., even in a larger group, the node peer-syncs with at most 2 neighbors, and the message send buffer is bounded to 100 messages, in order to adjust the link capacity settings to a contemporary transmission technology widely used. *Unlimited* ($\Lambda_{\max}^{\infty}$) link settings allow all possible links to be used for peer syncs, and the send buffer and the nodes' storages are unbounded. The Unlimited link settings are chosen to test a hypothetical upper bound for a possible message spread under conditions unconstrained by a given transmission technology and local storage. Note that the local storage is only unbounded for a link setting of $\Lambda_{\max}^{\infty}$, otherwise it is at 10,000 messages as stated above. Also, the sync strategies Random and PR do not have any effect for $\Lambda_{\max}^{\infty}$ syncing results.

B. Simulation Results

To evaluate the effectiveness of the microblogging system, the global message spread metric σ is used, and it is defined as

$$\sigma = \frac{\sum_{m \in M} \frac{rec_m}{|group_m|}}{|M|},$$

where M is the set of messages whose spread is monitored, rec_m is the number of group members (including sender) who received the message m, and $|group_m|$ is the number of group members belonging to the group under which message the m has been encrypted. Consequently, if all group members receive a message m, then its spread is 1 (full spread), and if all messages are fully spread, the global message spread is 1. The results are first presented and analyzed for the empirical Congress, Lausanne and Shanghai mobility inputs, and then for the synthetic datasets.

1) Empirical Mobility Data Results: The results of the Congress1 (C1) and the Congress2 (C2) datasets are depicted in Figures 5(a), (b). They show the message spread over time for three different sync strategies with error bars for selected points in time using Bluetooth ($\Lambda_2^{0.1k}$) link settings and the upper bound obtained when unlimited ($\Lambda_{\max}^{\infty}$) link settings are applied. The smaller C1 sample reaches close to 50% global message spread after 500 rounds, the larger C2 sample is slightly above 30% for random syncing. With $PR_{0.4}$ syncing, both datasets reach 75% message spread, because the prioritized selection of messages yields a better propagation. Even though the standard deviation is large, PR peer syncing runs with Bluetooth settings converge to the unlimited link setting results by a margin of approx. 15%–20%, whereas RD peer syncing performs much worse.

The results for the Lausanne1 and Lausanne2 datasets are shown in Figures 5(c), (d): with Bluetooth settings, the Random, $PR_{0.4}$ and $PR_{1.0}$ syncing methods yield final global message spreads slightly below 10%, and the spread with unlimited settings reaches around 55%, even though the number of nodes is similar to that of the C2 dataset. These results are due to the large area covered by the nodes and the bipartitioned network, whose two node clusters exhibit little message exchanges, as analyzed in Sect. III.

For the Shanghai datasets seen in Figures 5(e),(f) the results also show spreads between 10%–18% for the syncing with Bluetooth settings and are thus similar to the results obtained from the Lausanne datasets, despite the fact that over 4000 nodes participate in the network. The results for $\Lambda_{\max}^{\infty}$ syncing reach 95% final spread, with a steep increase in the first 50 rounds to 70% for Shanghai1 and 60% for Shanghai2. This is due to the high connectivity of the Shanghai datasets with about 9-10 times higher number of meetings per node compared to that of the Congress and Lausanne datasets.

Concluding, it can be stated that mobile microblogging, using solely local point-to-point links works well for sufficiently connected scenarios such as Congress1 and Congress2, with link settings adjusted to Bluetooth. In the Lausanne datasets the bipartition of the network hinders better message spread, and in the Shanghai datasets the transmission technology's bandwidth is the limiting factor. This shows that the system works for small node numbers, and that it could be deployed for larger ones if communication technologies with sufficient bandwidth were available.

2) Synthetic Mobility Data Results: We now review the ManhattanGrid and GaussMarkov synthetic datasets results. The results after 500 rounds of simulation with Bluetooth link settings for $PR_{0.4}$ and Random syncing are shown in Figures 6(a), (c) for ManhattanGrid mobility. For GaussMarkov mobility, this is shown in Figures 6(b), (d), respectively. Each figure depicts a heat map of the final global message spread σ under a fixed sync strategy, for mobility inputs with increasing number of nodes on the x-axis and increasing area on the y-axis.

Random syncing for ManhattanGrid achieves a message spread of 70% for 25 nodes up to an area of $25km^2$, and for 200 nodes for areas up to $5km^2$. GaussMarkov only achieves this message spread for 25 nodes up to $5km^2$, and for 100 nodes up to $3km^2$. $PR_{0.4}$ shows higher message spreads: 70% spread is reached for 350 nodes up to $10km^2$ in ManhattanGrid, and in GaussMarkov for 125 nodes up to $5km^2$. GaussMarkov obtains a message spread close to 100% only for small areas and 25 nodes, whereas ManhattanGrid exhibits such a spread up to areas of $15km^2$ for 25 nodes, which then decreases to areas of up to $5km^2$ for 150 nodes. This shows again that the microblogging system works in smaller areas under Bluetooth link settings for moderately sized node numbers, but does not scale well for higher node numbers in these areas.

The more densely connected mobility of ManhattanGrid leads to a higher message spread than the less structured mobility of GaussMarkov, and the selfish $PR_{0.4}$ peer-syncing similarly improves message spread over the Random peer-syncing. Overall, the results are counter-intuitive: the observation of a degressive spread trend for increased node numbers within the same area, and thus higher node connectivity that

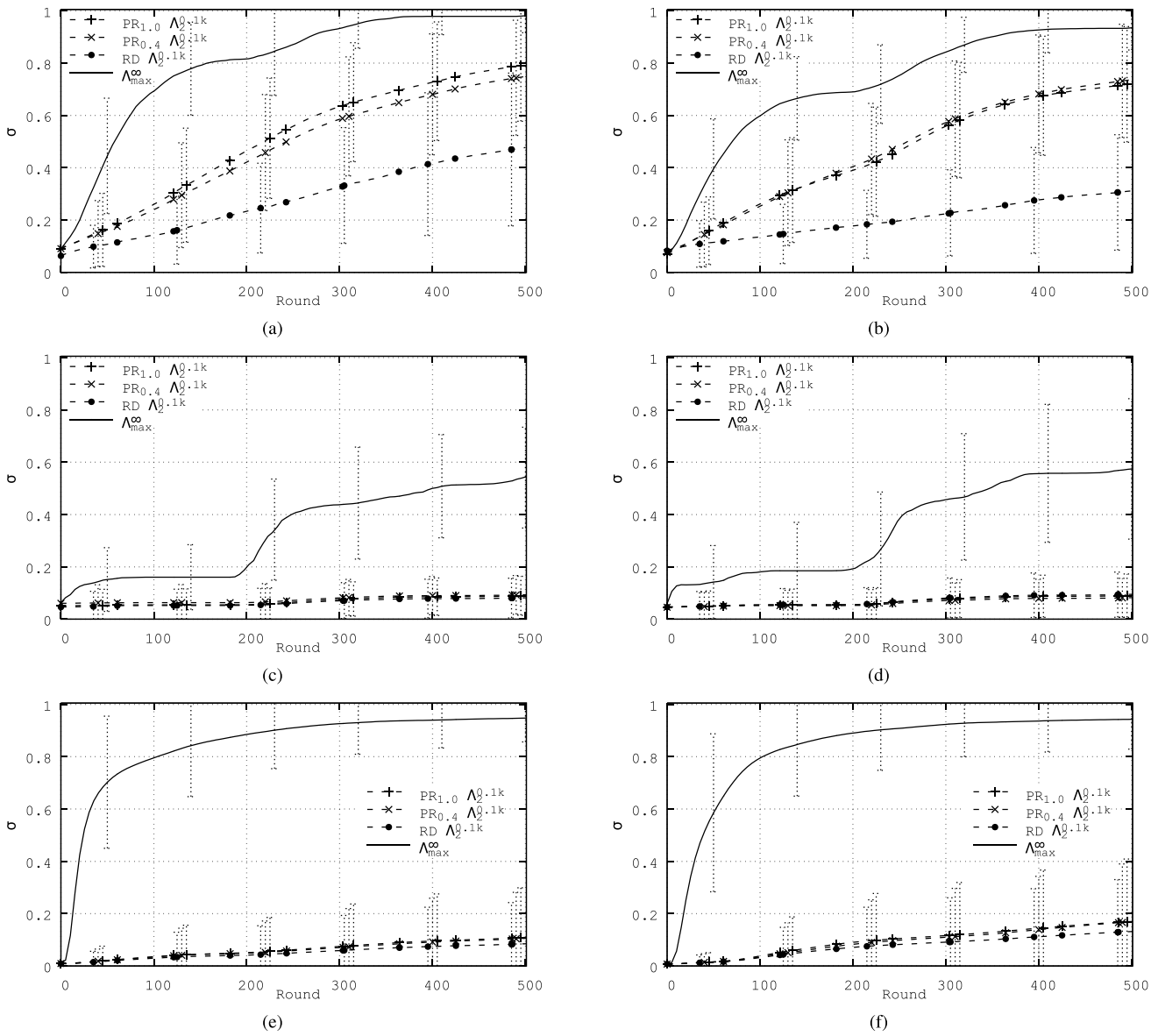


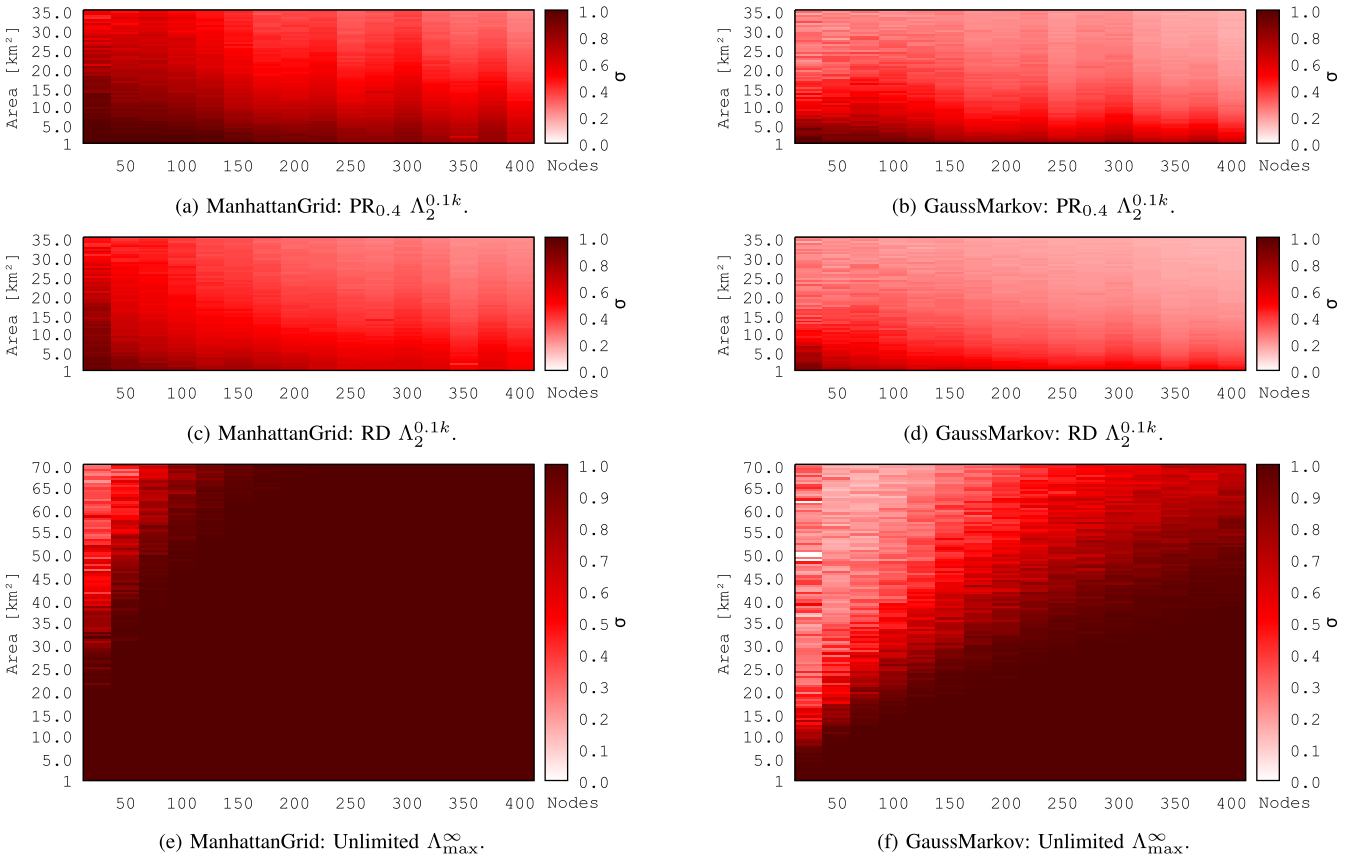
Fig. 5. Empirical mobility: global message spread σ with Bluetooth link settings for Prioritized ($\text{PR}_{0.4}\Lambda_2^{0.1k}$, $\text{PR}_{1.0}\Lambda_2^{0.1k}$) and Random (RD $\Lambda_2^{0.1k}$) syncing, and unlimited link settings for $\Lambda_{\max}^{\infty}$ syncing. (a) Congress1. (b) Congress2. (c) Lausanne1. (d) Lausanne2. (e) Shanghai1. (f) Shanghai2.

does not yield a better message spread, is a sign for bandwidth as limiting factor, beside the mobility of the nodes.

Under unlimited conditions the final results after 500 rounds of simulation can be seen in Figure 6(e) for ManhattanGrid and in Figure 6(f) for GaussMarkov. The propagation of the messages is positively influenced for higher node numbers in the same area as anticipated, resulting in a progressive message spread trend. A spread close to 100% is reached in ManhattenGrid for 25 nodes up to areas of $20km^2$, and it is constantly increasing so that from 200 nodes on, full spread is achieved in areas greater than $70km^2$. In GaussMarkov nearby 100% spread is reached in areas up to $7.5km^2$ for 25 nodes, and from 400 nodes on, full spread is reached in areas greater than $50km^2$.

To approximate how well link settings other than the introduced Bluetooth setting can converge to the Unlimited setting, that is, the hypothetical upper bound, we ran a number of experiments with varied link settings. Figures 7(a),(b) show the global message spread mean $\bar{\sigma}$ over time for the synthetic mobilities, defined as $\bar{\sigma} = \frac{\Sigma_{n\in N}\Sigma_{a\in A}\sigma_{n,a}}{|N||A|}$, with $N = \{25, 50, \ldots, 400\}$ being the set of node numbers, $A = \{1, 1.5, \ldots, 70\}$ the set of areas, and $\sigma_{n,a}$ the global message spread for a scenario with n nodes in an area of a km^2. For ManhattanGrid in Figure 7(a), the square-shaped points at round 500, named 6(a), (c), (e), correspond to the $\bar{\sigma}$-values of the heat map figures of the same name. For GaussMarkov in Figure 7(b), the square-shaped points analogously correspond to the Figures 6(b), (d), (f).

Overall we see that, for $\text{PR}_{0.4}$ peer syncing, the hypothetical upper bound can be reached by a margin of approx. 2% for ManhattanGrid with a $\Lambda_{\max}^{10k}$ link settings (and a margin of approx. 6% with a $\Lambda_2^{2.5k}$ link setting). For GaussMarkov,



(a) ManhattanGrid: $\mathrm{PR}_{0.4}\ \Lambda_2^{0.1k}$.

(b) GaussMarkov: $\mathrm{PR}_{0.4}\ \Lambda_2^{0.1k}$.

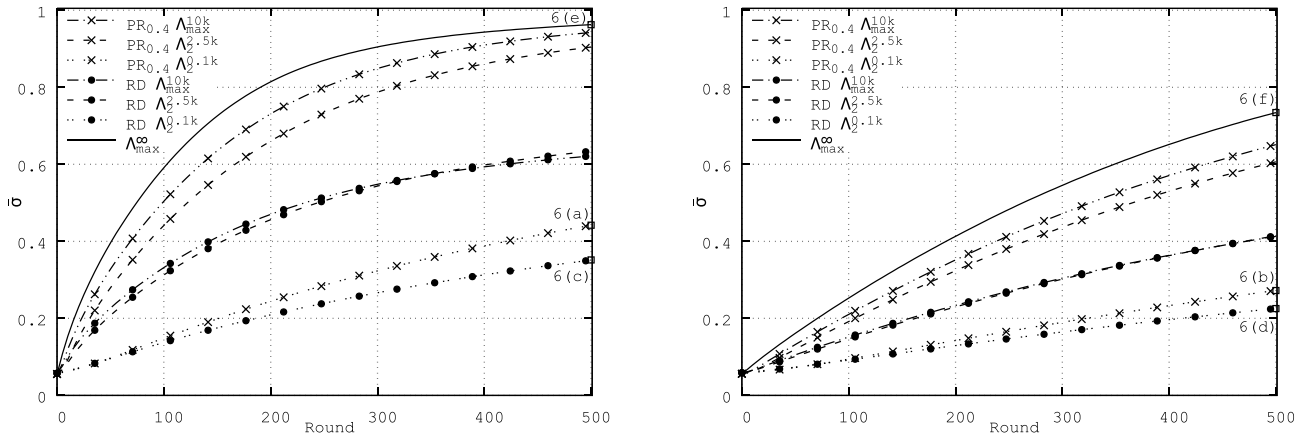
(c) ManhattanGrid: RD $\Lambda_2^{0.1k}$.

(d) GaussMarkov: RD $\Lambda_2^{0.1k}$.

(e) ManhattanGrid: Unlimited $\Lambda_{\max}^{\infty}$.

(f) GaussMarkov: Unlimited $\Lambda_{\max}^{\infty}$.

Fig. 6. Synthetic mobility: final round global message spread σ with Bluetooth link settings in (a)–(d) for Prioritized ($\mathrm{PR}_{0.4}\Lambda_2^{0.1k}$) and Random (RD $\Lambda_2^{0.1k}$) syncing, and unlimited link settings in (e)–(f) for $\Lambda_{\max}^{\infty}$ syncing.



(a)

(b)

Fig. 7. Synthetic mobility: mean global message spread $\bar{\sigma}$ with varied link settings for Prioritized (PR) and Random (RD) syncing and unlimited $\Lambda_{\max}^{\infty}$ link settings. The square-shaped points at round 500 named 6(a), (c), (e) and 6(b), (d), (f) show the $\bar{\sigma}$-values of the correspondingly captioned heat map figures for ManhattanGrid and GaussMarkov, respectively. (a) ManhattanGrid. (b) GaussMarkov.

the margins are about 8% and 12%, respectively. For Random (RD) peer syncing, the hypothetical upper bound cannot be reached for both mobilities, and an increase in link capacity from $\Lambda_2^{2.5k}$ to $\Lambda_{\max}^{10k}$ only yields a marginally better spread (and for ManhattanGrid the final values with $\Lambda_{\max}^{10k}$ link settings are slightly lower than with $\Lambda_2^{2.5k}$ settings). Random syncing results in nodes that indiscriminately re-send duplicates of the same messages. For a higher message spread, nodes should therefore use Prioritized syncing.

In conclusion, it can be stated that the limiting factor for the propagation is the capacity of the transmission channel, and not the empirical/synthetic mobility experienced by the nodes. Prioritizing messages always yields a better message spread than random syncing. Peer-syncing with more

TABLE III
COMPARISON OF EMPIRICAL AND SYNTHETIC SPREAD AFTER 500 ROUNDS

Scenario	Avg. node degree	Node count	$PR_{0.4}\ \Lambda_2^{0.1k}$	$RD\ \Lambda_2^{0.1k}$	$\Lambda_{\max}^{\infty}$
Congress1	1.19	40	75%	48%	98%
ManhattanGrid-50-19	1.17	50	72%	55%	100%
GaussMarkov-50-6	1.10	50	66%	53%	100%
Congress2	1.39	162	74%	31%	93%
ManhattanGrid-150-40	1.39	150	38%	31%	100%
GaussMarkov-150-15	1.38	150	37%	31%	100%
Lausanne1	2.49	141	11%	9%	57%
ManhattanGrid-150-25	2.43	150	46%	39%	100%
GaussMarkov-150-9	2.39	150	52%	40%	100%
Lausanne2	2.73	137	10%	9%	66%
ManhattanGrid-150-23	2.75	150	51%	41%	100%
GaussMarkov-150-8	2.63	150	52%	42%	100%

than 2 peers per round does only yield better spread for datasets with node degrees higher than 2, i.e. the Shanghai datasets.

3) Relation Between Empirical and Synthetic Mobility Results: The real-world datasets represent specific scenarios which correspond in part to some configurations of synthetic mobility models. In this section, we draw a comparison between the Congress and Lausanne datasets and their synthetic counterparts. Due to their high number of nodes, we do not analyze synthetic scenarios corresponding to the Shanghai datasets.

To compare results, we selected the synthetic scenarios exhibiting the closest node count and 2-hour average node degree as the corresponding real-world dataset.

Table III presents the spread after 500 rounds with different synchronization strategies. The synthetic scenarios are labeled with their mobility model, followed by node count and area in km^2.

We see that there is little spread discrepancy in the Congress datasets, and almost nonexistent for the $RD\ \Lambda_2^{0.1k}$ message synchronization strategy. However, Lausanne1 and Lausanne2 systematically show worse performance than the corresponding synthetic scenarios due to the network graph being nearly bipartite, with only a few messages being exchanged between the main two groups of nodes, which demonstrates the impact of social aspects on performance.

V. PRIVACY AND SECURITY

In this section we evaluate the goals of confidentiality and anonymity. They are assessed against a global passive adversary monitoring all network communication (peer syncs). The goal of censorship-resistance is evaluated in the presence of nodes either jamming the system or sending spam.

A. Confidentiality and Anonymity

The global passive adversary is not able to read any of the encrypted messages, thus keeping confidentiality. Receiver anonymity is achieved, since message content and group memberships remain confidential, and thus the adversary can only hypothesize about senders and receivers.

For sender anonymity, a global passive adversary monitors all peer syncs taking place, and retrospectively tries to identify the possible sender sets, assuming that a message was created a specific number of rounds ago. The subsequently discussed figures are based on a link setting with two peer syncs to show the achievable sender anonymity levels.

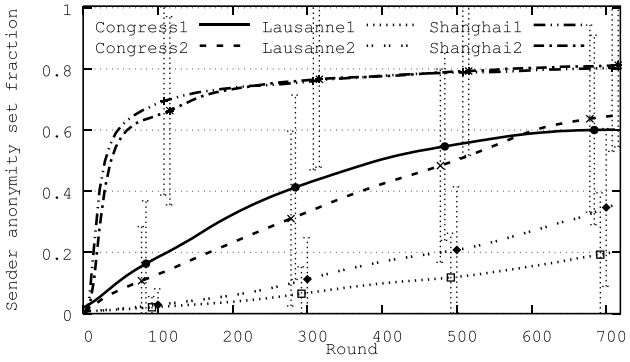


Fig. 8. Sender anonymity over time for empirical mobility: given that a received message has been created x rounds beforehand, what fraction of the total node number on average could have been the sender.

Figure 8 depicts sender anonymity over time for the empirical datasets. Sender anonymity set fraction signifies how many nodes on average could have sent a message to a receiving node, given that this message has been created x rounds beforehand, e.g., for Shanghai2, 60% of the 4445 nodes could have been the sender after 50 rounds, whereas for Lausanne2, only 20% of the 137 nodes could have been the sender after 700 rounds. Generally, we can observe that the underlying mobility highly determines how fast sender anonymity increases. The high connectivity of the Shanghai datasets thus yields a large possible anonymity set fraction after few rounds, and the Lausanne datasets—due to their bi-partitioned nature—yield small anonymity set fractions, even after 700 rounds.

In Figures 9(a) and 9(b), for the synthetic datasets, the number of rounds needed to achieve sender anonymity set sizes of 80% with respect to the total node number are shown as a heat map; no color indicates levels of anonymity unobtainable within the illustrated upper bound of rounds, but possibly with a larger one. The more connected the nodes are, the faster a certain anonymity level can be reached.

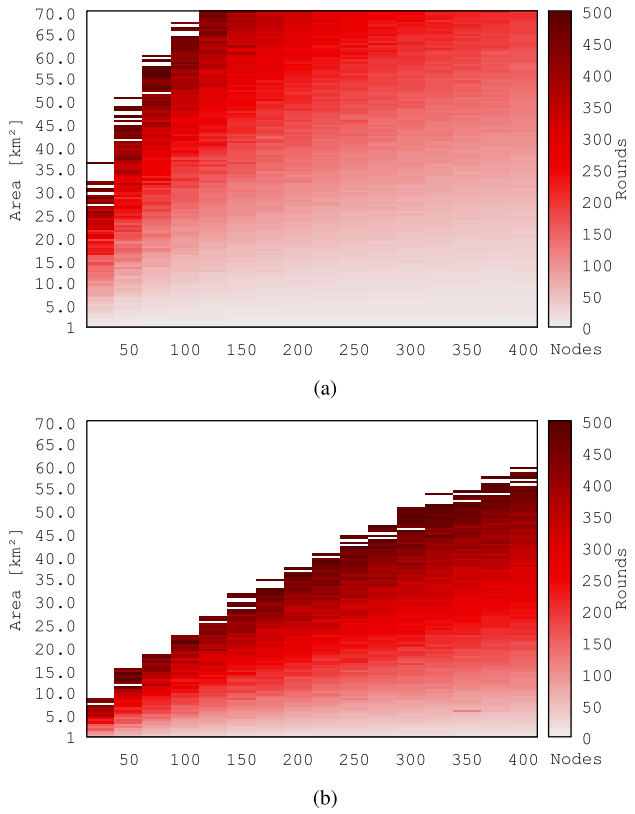


Fig. 9. Sender anonymity for synthetic mobility: the color indicates how many rounds ago a received message would have had to be created, so that 80% of the total node number on average could have been the sender—no color means this level of anonymity can not be reached in the given scenario within the observed time span. (a) ManhattanGrid. (b) GaussMarkov.

Overall, the achievable sender anonymity is closely linked to the underlying mobility.

B. Censorship-Resistance

Censorship-resistance means that an active adversary is unable to stop the propagation of messages based on their content, and that the system is not rendered dysfunctional in terms of the achieved message spread. To evaluate censorship-resistance, we assume an adversary that either jams a fraction of the nodes per round or one that injects spam messages.

Jamming is an attack used to stop the peer-syncing between nodes by creating interference. We denote jammed simulation runs by J_j, with $0 \leq j \leq 1$ being the fraction of all nodes which are each round randomly disabled from peer-syncing.

Selected message spread results with $J_{0.2}$, $J_{0.5}$ and $J_{0.8}$ simulation runs are shown in Figure 10(a) for the Congress2 empirical mobility, and in Figure 10(b) for the ManhattanGrid synthetic mobility. The final round message spread of Fig. 10(a) of $J_{0.2}$ is about 22% lower than the unlimited ($\Lambda^{\infty}_{\max}$) one. The subsequent final round differences amount to approx. 8% and 10% for $J_{0.5}$ and $J_{0.8}$, respectively, totaling to a loss of around 40% in the $J_{0.8}$ with respect to the $\Lambda^{\infty}_{\max}$ run. The synthetic ManhattanGrid results shown in Fig. 10(b) exhibit a final round loss of approx. 2% from its unlimited run, the subsequent losses being around 5% and 15%, amounting to a 22% loss in $J_{0.8}$ with respect to $\Lambda^{\infty}_{\max}$. We can see

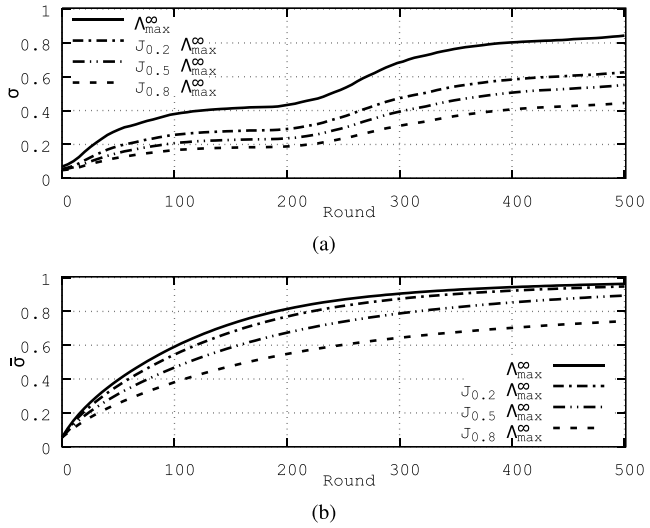


Fig. 10. Jamming: selected jammed runs with unlimited ($\Lambda^{\infty}_{\max}$) syncing, with Fig. 10(a) showing global message spread results for Congress2 and Fig. 10(b) mean global message spread results for ManhattanGrid mobility. (a) Empirical Congress2 mobility. (b) Synthetic ManhattanGrid mobility.

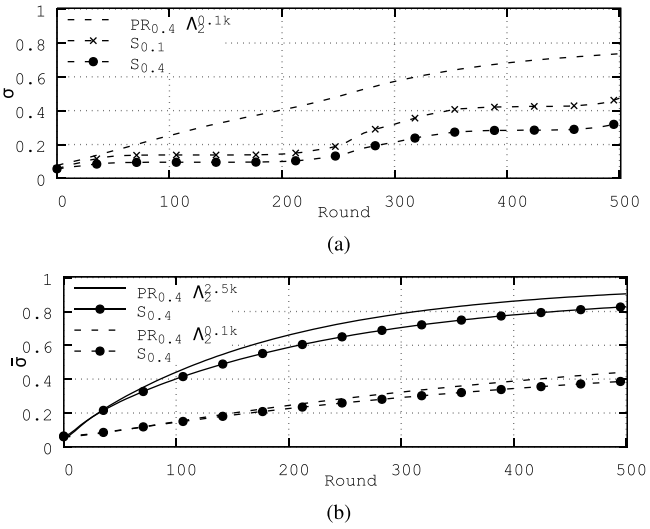


Fig. 11. Spamming: selected spam runs with Prioritized$_{0.4}$ (PR$_{0.4}$) syncing, with Fig. 11(a) showing global message spread results for Congress2 and Fig. 11(b) mean global message spread results for ManhattanGrid mobility. (a) Empirical Congress2 mobility. (b) Synthetic ManhattanGrid mobility.

that jamming significantly lowers the spread and that the Congress2 mobility is more susceptible to spamming. Yet, the microblogging system still exhibits sufficient message propagation.

Spamming is the injection of superfluous garbage messages into the network. The spamming nodes' intent is a Denial-of-Service attack on the microblogging system, since their bogus messages can not easily be filtered out and create network load. We denote spamming by S_s, with $0 \leq s \leq 1$ being the fraction of nodes that are spammers. All spamming nodes are not a member of any group and they peer-sync by filling their send buffers with new spam messages each round. These messages are indecipherable under any group key. Selected message spread results are shown in Figure 11(a) for

the Congress2 empirical mobility, and in Figure 11(b) for the ManhattanGrid synthetic mobility. The Congress2 final round message spread in Fig. 11(a) is reduced from 74% to 47% when the regular $PR_{0.4}$ $\Lambda_2^{0.1k}$ is run with $S_{0.1}$ spamming (and reduced to 33% with $S_{0.4}$ spamming). The synthetic final round results in Fig. 11(b) only show losses of 8% and 5% with $S_{0.4}$ spamming in comparison to the regular $PR_{0.4}$ $\Lambda_2^{2.5k}$ and $PR_{0.4}$ $\Lambda_2^{0.1k}$ runs, respectively. If Random were used as sync strategy, the resulting message spreads would be lower.

In summary, the results show that such a system can deal with a certain amount of jamming and that it has some robustness to spamming, if Prioritized is used as sync strategy.

VI. Discussion

Since the microblogging system draws primarily on node mobility and on the technology used for peer-syncing, we address in this section some limitations these technologies impose, and briefly outline how these shortcomings could be overcome.

A. Limitations of Mobile Anonymous Microblogging

In a decentralized system, the message propagation is based on the mobility of the nodes in combination with the deployed wireless communication technology. This also implies that from a certain sparsity on in rural areas the scheme will cease to work. If one group has several clusters of member nodes moving in disjoint network partitions, their messages will remain isolated within their respective partitions. The Lausanne dataset exhibits a case with a possible message spread of around 50% due to an underlying bi-partitioned network structure. In such cases more node movement, determined by the underlying movement structures, or technologies that can conduct peer syncs over a larger distance need to be deployed to bridge partitions. However, one could also argue that the assignment to the groups in the Lausanne case was done without taking into account potentially existing social structures reflected in the movement, so that the resulting message spread was lower. Consequently, we consider the system to work better in small city boroughs, cf. ManhattanGrid and the Congress cases, and in scenarios with similar node movement patterns, i.e., the Shanghai cases.

Another potential system limitation stems from battery power. The most consuming operation in terms of battery are the cryptographic and the networking ones. To test the cryptographic computational needs, we developed a small Android application that decrypts 100 messages, i.e., the amount of messages exchanged during a peer sync. Tests on a Samsung S3 smartphone required around 1200 mW with a CPU load of up to 40%. The decryption took 1.55 seconds. Networking power levels depend on the transfer technology used to establish pairings between nodes, with a total power requirement increasing by 750 mW for Bluetooth and by 2500 mW for WiFi. With Bluetooth, the application could approximately run for 30 hours [1].

B. Existing and Upcoming Technologies

The Congress datasets are the empirical mobility that was collected in a field experiment using an app on Bluetooth-enabled Android-smartphones with local point-to-point communications as foreseen in the system. In this setting, longer pairing times were no obstacle, however, the pairing time of devices before they can peer-sync should ideally be lower than a couple of seconds, cf. the Shanghai datasets with its many contact durations below 1 second.

Besides pairing time, partitioned networks and scalability are issues asking for new technologies. Ideally, connection establishments should be shorter than 5 seconds, so that short contact durations could be used for peer syncs, paired with a greater link capacity. Then a maximum communication distance of $15m$, the baseline in all conducted simulations, might be sufficient, even though longer distances are desirable to better bridge partitions. Existing and upcoming communication technologies, both wireless and optical, that can be used for point-to-point peer-syncing, already exist [13]. One prospective optical technology that potentially could be deployed is the one presented in [14]. This optical technology, for instance, requires a free line-of-sight, what contrasts with radio-based technologies, which are susceptible to jamming.

VII. Related Work

We outline related work in general, then we focus on networking and mobility, and finish with microblogging systems.

An early peer-to-peer approach for anonymity in web transactions is Crowds [15]. Peers are used to create a mix cascade over which web transactions are routed, with transport encryption established between each pair of nodes. Mix networks [16] relay messages over multiple mixes that pool and send them over to the next mix until they reach their receivers. The two approaches of mix cascades and peer-to-peer have been extensively discussed in [17]. Onion routing [18] adds layered encryption to the mix network, and is the basis of Tor [19]. Both approaches, crowds and mixes, contain specific (uni-)cast routing. In contrast, we deploy universal re-encryption instead of a layered one, and messages spread agnostic to routing by using direct communications only (peer syncs).

A. Anonymous Wireless Networking

Several routing protocols aim at providing anonymity to users of wireless networks with an onion routing approach.

ANODR [20] builds a path between sender and receiver by flooding the network with route requests. Only the intended recipient is able to understand and reply to the request. The nodes on the reply path add encryption information to the route response so that the sender can build an onion around the data it wishes to transmit anonymously. ASR [21] and AnonDSR [22] work similarly. We note that a global adversary may correlate sender and recipient by observing who replies to which route request. The strength of onion routing on the Internet relies on the difficulty of observing the entire network, which may be less of an obstacle in wireless networks.

SDAR [23] hides the route requests and responses by making the nodes to exchange keys only with their 1-hop neighbors. Nodes assign their nearest neighbors a trust level according to their network behavior and share a group key with the ones in the same trust level to encrypt their communication. A disadvantage in this scheme is that the recipient will know

TABLE IV
OVERVIEW OF GOALS AND RESULTS

Goals	Results	Section
Network performance	- high message spread in well-connected networks, even with few nodes - prioritizing messages yields better message spread than random syncing - the capacity of the transmission channel is the limiting factor for propagation, regardless of movement patterns	IV-B
Confidentiality	- confidentiality is always guaranteed independently of the network characteristics	V-A
Anonymity	- higher anonymity levels are reached faster in well-connected networks - size of sender anonymity set strongly depends on user mobility	V-A
Censorship-resistance	- system can cope with some level of jamming - system is strong against spamming when using a syncing strategy that considers message prioritization	V-B

the identity of all nodes in the path. Additionally, malicious nodes might manipulate trust levels. In MASK [24], nodes also exchange keys with their nearest neighbors to hide the path of the request, however, the identity of the recipient is transmitted in plaintext in the route request.

ARM [25] tackles the disadvantages of the previously mentioned systems and proposes a solution based on single-use pseudonyms to build routes between sender and recipient.

The previous approaches work only if the path between sender and recipient is stable during the entire process. This is hardly the case in a network of mobile users. While not tackling anonymity, the authors of HumaNets [26] present a delay-tolerant approach to route messages between mobile nodes while protecting the users' location privacy. Such a network enables anonymization overlays to be deployed in highly mobile scenarios.

In [27] a way of anonymously accessing online services via the cellular network using hybrid networks is proposed. Mobile devices cooperate over a local WiFi network to provide anonymity to peers accessing online services, and peers are anonymously rewarded for cooperating via a micropayment scheme. In [28], a protocol made for smartphone users to anonymously communicate in a mobile cloud is presented. The protocol relies on opportunistic ad hoc communications between smartphones and social-network properties to provide anonymity. Other approaches to anonymity and security include [29]–[31].

B. Microblogging

Privacy-enhanced client-server microblogging similar to Twitter is proposed in [32], detailing a microblogging server that matches encrypted messages to subscribers using oblivious matching. However, as in [33], anonymity is no design goal. To achieve it, the use of Tor or a subscription service with unlinkability of user across logins such as [34] is needed.

Some wide area network dependent peer-to-peer microblogging systems exist, but they do not focus on anonymity and censorship-resistance. Fault-tolerance and decentralization are treated in [35]–[37], but not anonymity. In [38] a cryptographic mechanism allows a peer to anonymously request messages from peers, and in [39] a mix overlay network is constructed over existing Twitter users for anonymity. Yet, both approaches need Internet access, in contrast to our scenario.

Few microblogging systems in delay-tolerant networks as used in our system exist. Floating content [40] foresees information to stay geostationary in a fixed urban space, an idea for which criticality conditions have been derived [41]. In a network setting similar to ours, [2] proposes community reputation to prevent flooding of the network with junk messages. Yet, the foci are not on anonymity and censorship-resistance, and no other solutions in this niche are known to the authors.

VIII. CONCLUSION

We show how a mobile distributed microblogging system allows to spread messages by direct, proximity-based peer-syncing in real-world conditions. The available technologies pose a bottleneck to message spread, for instance, Bluetooth alone achieves only sub-optimal message propagation. This demands for better suited technologies having both higher bandwidth and faster connection establishment. Prioritized message propagation is preferable to Random syncing. With prioritization, the system performs reasonably well under adversarial jamming and spamming, thus showing the desired censorship-resistance, whilst maintaining message confidentiality and anonymity.

Table IV presents an overview of our goals, the corresponding results, and where they are analyzed in this work.

REFERENCES

[1] M. Senftleben, M. Bucicoiu, E. Tews, F. Armknecht, S. Katzenbeisser, and A.-R. Sadeghi, "MoP-2-MoP—Mobile private microblogging," in *Proc. 18th Int. Conf. Financial Cryptogr. Data Secur.*, 2014, pp. 384–396.

[2] Y. Ben-David and A. Kim. (2012). *Robin: An Attack-Resilient Microblogging Service to Circumvent Government-Imposed Communication Blackouts*. [Online]. Available: http://www.eecs.berkeley.edu/~yahel/papers/

[3] J. Pouwelse, "Media without censorship (CensorFree) scenarios," Dpt. Softw. Comput. Technol., Delft Univ. Technol., Delft, The Netherlands, Tech. Rep., Jul. 2012. [Online]. Available: https://tools.ietf.org/html/draft-pouwelse-censorfree-scenarios-02

[4] P. Golle, M. Jakobsson, A. Juels, and P. Syverson, "Universal re-encryption for mixnets," in *Topics in Cryptology-CT-RSA* (Lecture Notes in Computer Science), vol. 2964. Berlin, Germany: Springer, 2004, pp. 163–178.

[5] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "BonnMotion: A mobility scenario generation and analysis tool," in *Proc. ICST*, 2010, pp. 51:1–51:10.

[6] J. Schejbal, "A real-world study of mobile peer-to-peer networking," M.S. thesis, Dept. Comput. Sci., Technische Univ. Darmstadt, Darmstadt, Germany, 2014. [Online]. Available: http://www.janschejbal.de/permanent/masterthesis

[7] J. K. Laurila *et al.*, "The mobile data challenge: Big data for mobile computing research," in *Proc. 10th Int. Conf. Pervasive Comput. Newcastle*, Jun. 2012.

[8] *SUVnet-Trace Data*, accessed on May 23, 2014. [Online]. Available: http:/wirelesslab.sjtu.edu.cn

[9] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, Sep. 2002.
[10] European Telecommunications Standards Institute (ETSI), "Universal mobile telecommunications system (UMTS); selection procedures for the choice of radio transmission technologies of the UMTS (UMTS 30.03 version 3.2.0)," Tech. Rep. TR 101 112, Apr. 1998.
[11] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "TwitterRank: Finding topic-sensitive influential twitterers," in *Proc. WSDM*, 2010, pp. 261–270.
[12] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, 2009.
[13] D. K. Borah, A. C. Boucouvalas, C. C. Davis, S. Hranilovic, and K. Yiannopoulos, "A review of communication-oriented optical wireless systems," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, pp. 1–28, Dec. 2012.
[14] A. M. Shah, S. S. Ara, G. Kitazumi, and M. Matsumoto, "WLC10-3: IrSimple modeling and performance evaluation for high-speed infrared communications," in *Proc. GLOBECOM*, Nov./Dec. 2006, pp. 1–6.
[15] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for Web transactions," *ACM Trans. Inf. Syst. Secur.*, vol. 1, no. 1, pp. 66–92, Nov. 1998.
[16] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–90, Feb. 1981.
[17] R. Böhme, G. Danezis, C. Díaz, S. Köpsell, and A. Pfitzmann, "On the PET workshop panel 'mix cascades versus peer-to-peer: Is one concept superior?" in *Proc. Privacy Enhancing Technol. Workshop (PETs)*, 2004, pp. 243–255.
[18] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Hiding routing information," in *Information Hiding* (Lecture Notes in Computer Science), vol. 1174. Berlin, Germany: Springer, 1996, pp. 137–150.
[19] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. USENIX Secur. Symp.*, 2004, p. 21.
[20] J. Kong and X. Hong, "ANODR: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in *Proc. MobiHoc*, 2003, pp. 291–302.
[21] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao, and R. Deng, "Anonymous secure routing in mobile ad-hoc networks," in *Proc. LCN*, Nov. 2004, pp. 102–108.
[22] R. Song, L. Korba, and G. Yee, "AnonDSR: Efficient anonymous dynamic source routing for mobile ad-hoc networks," in *Proc. 3rd ACM Workshop Secur. Ad Hoc Sensor Netw. (SASN)*, 2005, pp. 33–42.
[23] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba, "SDAR: A secure distributed anonymous routing protocol for wireless and mobile ad hoc networks," in *Proc. IEEE LCN*, Nov. 2004, pp. 618–624.
[24] Y. Zhang, W. Liu, and W. Lou, "Anonymous communications in mobile ad hoc networks," in *Proc. 24th IEEE INFOCOM*, vol. 3. Mar. 2005, pp. 1940–1951.
[25] S. Seys and B. Preneel, "ARM: Anonymous routing protocol for mobile ad hoc networks," *Int. J. Wireless Mobile Comput.*, vol. 3, no. 3, pp. 145–155, Oct. 2009.
[26] A. J. Aviv, M. Sherr, M. Blaze, and J. M. Smith, "Privacy-aware message exchanges for geographically routed human movement networks," in *Proc. 17th Eur. Symp. Comput. Secur. (ESORICS)*, 2012, pp. 181–198.
[27] C. A. Ardagna, S. Jajodia, P. Samarati, and A. Stavrou, "Providing users' anonymity in mobile hybrid networks," *ACM Trans. Internet Technol.*, vol. 12, no. 3, pp. 7:1–7:33, May 2013.
[28] C. A. Ardagna, M. Conti, M. Leone, and J. Stefa, "An anonymous end-to-end communication protocol for mobile cloud environments," *IEEE Trans. Services Comput.*, vol. 7, no. 3, pp. 373–386, Jul./Sep. 2014.
[29] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos, "AnonySense: A system for anonymous opportunistic sensing," *Pervasive Mobile Comput.*, vol. 7, no. 1, pp. 16–30, Feb. 2011.
[30] S. Fahl, M. Harbach, T. Muders, M. Smith, and U. Sander, "Helping Johnny 2.0 to encrypt his Facebook conversations," in *Proc. 8th Symp. Usable Privacy Secur. (SOUPS)*, New York, NY, USA, 2012, pp. 11:1–11:17.
[31] G. Portokalidis, P. Homburg, K. Anagnostakis, and H. Bos, "Paranoid Android: Versatile protection for smartphones," in *Proc. 26th Annu. Comput. Secur. Appl. Conf. (ACSAC)*, New York, NY, USA, 2010, pp. 347–356.
[32] E. De Cristofaro, C. Soriente, G. Tsudik, and A. Williams, "Hummingbird: Privacy at the time of Twitter," in *Proc. Int. Symp. Secur. Privacy (S&P)*, May 2012, pp. 285–299.
[33] I. Singh, M. Butkiewicz, H. V. Madhyastha, S. V. Krishnamurthy, and S. Addepalli, "Twitsper: Tweeting privately," *IEEE Security Privacy*, vol. 11, no. 3, pp. 46–50, May/Jun. 2013.
[34] M. Z. Lee, A. M. Dunn, J. Katz, B. Waters, and E. Witchel, "Anonpass: Practical anonymous subscriptions," in *Proc. Int. Symp. Secur. Privacy (S&P)*, 2013, pp. 319–333.
[35] P. S. Juste, D. Wolinsky, P. O. Boykin, and R. J. Figueiredo, "Litter: A lightweight peer-to-peer microblogging service," in *Proc. 3rd Int. Conf. Social Comput.*, Oct. 2011, pp. 900–903.
[36] P. S. Juste, H. Eom, K. Lee, and R. J. Figueiredo, "Enabling decentralized microblogging through P2PVPNs," in *Proc. Conf. Consum. Commun. Netw.*, Jan. 2013, pp. 323–328.
[37] T. Perfitt and B. Englert, "Megaphone: Fault tolerant, scalable, and trustworthy P2P microblogging," in *Proc. 5th Internet Web Appl. Services*, May 2010, pp. 469–477.
[38] R. Fernando, B. Bhargava, and M. Linderman, "Private anonymous messaging," in *Proc. Symp. Rel. Distrib. Syst. (SRDS)*, Oct. 2012, pp. 430–435.
[39] J. Daubert, L. Böck, P. Kikirasy, M. Mühlhäuser, and M. Fischer, "Twitterize: Anonymous micro-blogging," in *Proc. IEEE/ACS 11th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2014, pp. 817–823.
[40] J. Ott, E. Hyytiä, P. Lassila, T. Vaegs, and J. Kangasharju, "Floating content: Information sharing in urban areas," in *Proc. Pervasive Comput. Commun. (PerCom)*, Mar. 2011, pp. 136–146.
[41] J. Virtamo, E. Hyytiä, and P. Lassila, "Criticality condition for information floating with random walk of nodes," *Perform. Eval.*, vol. 70, no. 2, pp. 114–123, Feb. 2013.

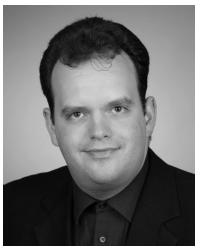
Marius Senftleben received a diploma in business computer science from the Technical University of Darmstadt in 2012. He is currently pursuing the Ph.D. degree in computer science with the Security Engineering Group, Technical University of Darmstadt. His research interests are in privacy-enhancing technologies, with particular focus on anonymity and privacy in online social networks.

Ana Barroso received the B.Sc. degree in computer engineering and information technology from the Instituto Superior Técnico, Lisbon, in 2009, and the M.Sc. degree in computer science (specialization in Internet and Web-based systems) from the Technical University of Darmstadt, in 2011, where she is currently pursuing the Ph.D. degree with the Secure Mobile Networking Laboratory. Her research interests are social, mobile, and infrastructure-independent secure wireless networks, with a special focus on user anonymity.

Mihai Bucicoiu was born in Pitesti, Romania, in 1986. He received the Ph.D. degree in computer science from the Politehnica University of Bucharest (UPB), in 2014. From 2009 to 2015, he was a Network Engineer for the Romanian Educational Network and has been the Coordinator of the Local Cisco Networking Academy. Since 2009, he has been a Teaching Assistant with the Computer Science Department, UPB. He has authored one book and more than 15 articles. His research interests include network security, operating systems security, usable security, and mobile security. He received the best paper award from ASIACCS in 2015.

Matthias Hollick received the Ph.D. degree from the Technical University of Darmstadt (TU Darmstadt), Germany, in 2004. He has been researching and teaching with TU Darmstadt, the Universidad Carlos III de Madrid (UC3M), and the University of Illinois at Urbana–Champaign. He heads the Secure Mobile Networking Laboratory with the Computer Science Department, TU Darmstadt. His research focus is on secure and quality-of-service-aware communication for mobile and wireless ad hoc, mesh, and sensor networks.

Stefan Katzenbeisser (S'98–A'01–M'07–SM'12) received the Ph.D. degree from the Vienna University of Technology, Austria. After working as a Research Scientist with the Technical University of Munich, Germany, he joined Philips Research as a Senior Scientist in 2006. Since 2008, he has been a Professor with the Technical University of Darmstadt, heading the Security Engineering Group. His current research interests include digital rights management, data privacy, software security, and cryptographic protocol design. He has authored more than 100 scientific publications and served on the program committees of several workshops and conferences devoted to information security. He was Program Chair of the Information Hiding in 2005, and General Chair of the ACM Conference on Computer and Communications Security in 2016. From 2009 to 2011, he served on the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society.



Erik Tews received the bachelor's, diploma, and Ph.D. degrees from the Technical University of Darmstadt, Germany, all in computer science. He was a Postdoctoral Researcher with the Technical University of Darmstadt in 2015. In 2015, he was a Lecturer with the University of Birmingham, U.K. His research interests include the security of wireless protocols, applied cryptanalysis, embedded systems, privacy preserving systems, the Internet of Things, and the security of railway applications.