Privacy and temporal aware allocation of data in Decentralized Online Social Networks

Andrea De Salve[1,2], Barbara Guidi[2], Paolo Mori[1],
Laura Ricci[2], and Vincenzo Ambriola[2]

[1] Istituto di Informatica e Telematica
Consiglio Nazionale delle Ricerche,
Via G. Moruzzi, 1 - Pisa - Italy,
[2] Department Of Computer Science - University of Pisa,
Largo Bruno Pontecorvo, Pisa - Italy
Email:desalve,guidi,ricci,ambriola@di.unipi.it
paolo.mori@iit.cnr.it

Abstract. Distributed Online Social Networks (DOSNs) have recently been proposed to grant users more control over the data they share with the other users. Indeed, in contrast to centralized Online Social Networks (such as Facebook), DOSNs are not based on centralized storage services, because the contents shared by the users are stored on the devices of the users themselves. One of the main challenges in a DOSN comes from guaranteeing availability of the users' contents when the data owner disconnects from the network. In this paper, we focus our attention on data availability by proposing a distributed allocation strategy which takes into account both the privacy policies defined on the contents and the availability patterns (online/offline) of the users in order to allocate their contents on trusted nodes. A linear predictor is used to model and to predict the availability status of the users in a future time interval, on the basis of their past temporal behaviour. We conduct a set of experiments on a set of traces taken from Facebook. The results prove the effectiveness of our approach by showing high availability of users' profiles.

Keywords: Decentralized Online Social Networks, Data Availability, Data Privacy, Availability Prediction

1 Introduction

Online Social Networks (OSNs) [12] have become crucial for communication and contents sharing between users of the Internet. According to different statistics[3] the use of OSN platforms has spread at an impressive rate across the world and users generate a high-volume of valuable information. As a result, the presence of a such large amount of personal information about users has created new business models.

[3] International Telecommunication Union (ITU) - Measuring the Information Society Report 2016 http://www.itu.int/en/ITU-D/Statistics/Documents/publications/misr2016/MISR2016-w4.pdf

Nowadays, popular OSNs are mainly based on centralized architectures and information shared by the users on the platform may be exploited for targeting advertisements or for other purposes. Such centralized architectures of current OSNs have raised several problems related to both privacy risks and performance issues [18, 13, 27].

Distributed Online Social Networks (DOSNs) [7] have been proposed as an alternative to the centralized solutions to allow a major control of the users over their own contents. A DOSN is made up of a (dynamic) set of peers, such as a network of trusted servers, a P2P system or an opportunistic network, which collaborate with each other in order to provide the social services. Even though DOSNs solve the problem of control over user contents, the decentralization of the OSN introduces new security issues related to the availability of user contents and privacy of those contents with respect the other users of the DOSNs. In fact, DOSNs allow their users to create different types of contents (such as posts, images, comments, likes, etc.) and to organize them under relevant categories in their profiles. In order to support fine-grained access control, current DOSNs enable users to regulate the access to the contents shared in their profile by using privacy policies. Such policies are statements that specify the users authorized to access the contents in terms of a set of features, modeled by the attributes (such friendship, friendship type, etc.).

An important challenge of DOSNs is to guarantee the availability of such contents even when the user who published them is offline. Replication is the most widely used technique to ensure data availability and it consists of storing multiple copies of the same content on several devices available in the system. In DOSNs, the classical replication approach consists in selecting random peers, where the contents should be stored, from the users' devices available in the network. Other replication approaches set specific performance objectives for the selection of replica peers (such as online-online correlation between time spent online by users [22], average time spent online by users [5],[11]). However, confidentiality of contents must be protected in order to avoid disclosure of information to unauthorized users that store the contents. Indeed, it is possible that the device selected as a replica peer for a content belongs to a user who is not authorized to access it. As a result, the solution typically adopted by existing DOSNs [6, 4, 1] consists in encrypting the contents before being stored on user devices, in such a way that only the users who have the access right can access them. Although this kind of design ensures high data availability, it suffers from performance issues due to the overhead introduced by the encryption mechanisms [8, 2].

In this paper, we propose a strategy for a privacy aware allocation of the contents to the peers of the DOSN which improves the results we presented in [11] because it ensures a high availability of such contents by exploiting the availability patterns of the users to define the content allocation. In particular, the allocation strategy we propose enhances the availability of contents by replicating them on peers which belong to users authorized to access them. In addition, the strategy also evaluates the availability patterns of the related users

in order to choose the devices where contents can be stored. For this purpose, we use a linear predictor which predicts the availability status of the users (i.e., online or offline) in a future time interval on the basis of their past behaviour. Summarizing, the proposed allocation strategy can effectively improve the content availability because each content is replicated on the peers of users who are both *i)* authorized to access it, and *ii)* most likely to be online for a period of time in the future. We conducted a number of simulations based on the real behavior of users extracted from a Facebook dataset, and we demonstrated that our allocation strategy ensures an higher level of contents availability compared to the classical approaches proposed by current DOSNs.

This paper is organized as follows. Section 2 reviews related work about privacy protection and data availability in DOSNs. Section 3 presents preliminary concepts on modeling OSNs contents. Section 4 discusses the general architecture of our system, while Section 5 introduces the new strategy we propose for replica management. Section 6 reports the experimental results. Finally, conclusions and future work are presented in Section 7.

2 Related Work

In this section, we provide an overview of current DOSN's approaches used to enforce privacy control over users' data. Current DOSNs use simple privacy policies by coupling distributed approaches with encryption techniques, mostly the Public Key Infrastructure (PKI) and the Attribute Based Encryption (ABE).

The first proposed DOSN is Diaspora[4], where users are able to act as local servers in order to keep complete control over their data. In Safebook [6] users can specify, for each friend, a trust level used to select closely related contacts that will store user's data. A similar approach is proposed also by My3 [22], where users are able to choose a set of trusted friends where the contents can be stored. Authors of [19] propose to store data on the set of trusted friends, chosen by using the Dunbar's approach. PeerSoN [4] exploits local users devices to store their data securely. Users data are encrypted with the public keys of the users who have access to it. LotusNet [1] exploits a combination of both symmetric and asymmetric encryption. In SuperNova [26] users data may be stored on untrusted peers or on users friends peers. The data stored on untrusted peers (or on friend peers) are encrypted by using a threshold-based secret sharing approach. LifeSocial.KOM [17] uses a DHT to store and replicate users' data. Then, this symmetric key is encrypted individually with the public key of the users able to access the data and attached to the users content. In Cachet [24] users data are securely stored and replicated on the peers of a DHT by using a cryptography hybrid structure which leverages ABE and symmetric key. Only users that meet the policy can decrypt the private key used to encrypt the content. In SOUP [21] users can store private data on other participants without the usage of a DHT by creating mirror replicas and every replica must be encrypted. Authors

[4] http://joindiaspora.com

of [25] build a replication strategy by considering the underlying social graph and they evaluate two types of availability: pure and friend availability. Other possible selection strategies and a comparison between them have been proposed in [5] and [22].

3 Modeling Social Profiles

In our system, the social profile P_u of a user u is modeled by a tree whose nodes correspond to the contents belonging to P_u. The structure of the information contained in the user's profile (images, albums, posts, comments, likes, blogs, pages, etc.) is well suited to be represented by using a hierarchical data structure, such as a tree. A detailed explanation of the hierarchical profile structure we defined is reported in [10]. The root of the tree is considered as the entry point for all the contents related to a profile owner. Each node of the profile tree embeds information about the identifiers of the children and parent nodes and is paired with the privacy preferences chosen by the owner. Note that, since we have a one to one mapping between the nodes of the tree P_u and the user's contents, we use interchangeably the terms node or content to refer them. Furthermore, we suppose that an unique identifier is assigned to each node of the user's content tree.

To achieve fine-grained access control, users can specify the authorization on the contents of their profile by defining privacy policies. The privacy policies defined by a user for his contents are collected in the *privacy policy repository* of the user.

We exploit the authorization framework proposed in [10] to support users in the management of their privacy preferences. The framework allows its users to define flexible privacy policies to regulate the accesses to the content they have shared by means of a proper Privacy Policy Language.

4 System Architecture

This section summarizes the general architecture of our system, which has been proposed in [11]. We assume a one-to-one mapping between users and their peers and we use interchangeably the terms peer or user to refer them.

The system we propose is based on a multi-layered architecture, exploiting both a DHT and direct communications between the peers. The profile of a user is replicated on a set of peer, as described in the following.

Each user u is bound to its user descriptor D_u which contains information about the IP address of the corresponding peer, the online/offline status of the peer, the identifier of the root of its profile tree and the current replicas available for all contents of the profile. Since the descriptor D_u must be available to all the peers which are going to access the profile of u when it is offline, it is stored on a DHT and may be retrieved by exploiting the identifier of the u. The DHT provides a secure storage layer and look-up service which is used to store information required to identify users and their devices (such as the current IP

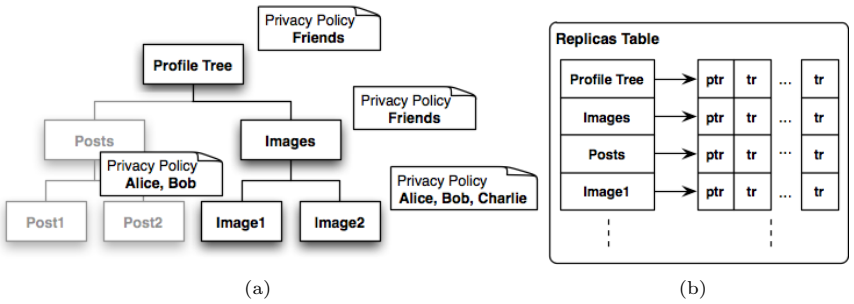


Fig. 1. Fig. 1(a) shows the general profile tree data structure which contains posts and images of the user U. The root of the profile tree and the *Images* node are intended to be shared with the entire circle of U's friends. Finally, the privacy policies specified by U for the contents *Image1* and *Image2* allow access to the set of users $\{Alice, Bob\}$ and $\{Alice, Bob, Charlie\}$, respectively. While Fig. 1(b) shows the replicas table for the profile tree of U where names of the nodes are used as content ids. A primary trusted replica (ptr) and a set of further trusted replicas (tr) are defined for each node.

address of the user) and to find the replicas where contents are stored. Users and contents are identified by using the DHT identifier space. Note that the DHT is exploited not only to keep track of the content replicas, but it is also indispensable for peer bootstrapping, addressing and for supporting the search of new friends.

The association between the contents of a profile tree P_u and the trusted replica peers where they are stored are maintained in the *replicas table* R_u (see Fig. 1(b)) which is located in the user's descriptor D_u of the profile owner.

In order to ensure higher data availability, each content of a profile tree is replicated on k peers, where k is an input parameter of the system. The replicas table R_u provides information about the current trusted replicas available for the profile P_u and it contains, for each content c of P_u, the trusted replica list $R_u(c) = \{ptr, tr_1 \ldots, tr_{k-1}\}$, where ptr is the primary trusted replica, while $\{tr_1 \ldots, tr_{k-1}\}$ are the other replicas of the content. Let us denote by $tr(c)$ and $ptr(c)$ respectively, the set of trusted replicas and the primary trusted replica for the content c. For maintaining replication transparency, the trusted replica lists are managed according to a *passive replication model* [14] where every user communicates only with ptr, the *primary trusted replica*. When the profile owner is online, it becomes primary trusted replica of his contents. Primary trusted replica peers are responsible for the availability of the contents, for enforcing privacy policies every time a user tries to access a content, and for the selection of the new trusted replicas. Specifically, when a user v requests access to the content c of P_u, the replica storing c evaluates the privacy policies of u (linked to the profile P_u) in order to decide whether to permit or deny the access to c.
Only the primary replica can elect new replicas for the content and it can add them at the end of the trusted replica list. If the primary replica crashes abruptly or voluntary leaves the DOSN, the availability of their contents is guaranteed by

the other trusted replicas. The selection of a new trusted replica for a content c of user profile P_u can be performed: *i)* actively by the content owner u during the online periods since it acts as primary replicas for their contents or *ii)* by the current primary trusted replica of c, when the owner is not online. Indeed, when a user u becomes a primary trusted replica for a content c, it has to periodically check if new trusted replicas are needed and, in this case, it has to find another peer who is allowed to access c according to u's privacy policy. Robustness against peers failure and involuntary disconnections may be ensured through periodical exchanges of heartbeat messages between the primary replica and trusted replica peer. As a result, the amount of time data is kept available depends on both the effectiveness and complexity of the strategy with which the trusted replicas are chosen.

5 Replica Selection: our proposal

This paper proposes an enhancement of the framework we proposed in [10] exploiting a new replica selection strategy which, besides preserving the privacy preferences defined by the users, also guarantees a high level of data availability. In particular, the data allocation strategy exploits both online pattern of users and privacy policies defined by the users on their contents in order to decide on which user's device a content should be replicated. As proposed in our previous work [11], the privacy policy specified for the content c is used is used by the primary trusted replica to retrieve the set of users authorized to access the content c. The privacy policy is evaluated by using the authorization component of the privacy-preserving framework and by simulating an access to the content in order to define set of possible *trusted peers* to host replicas for a content c, i.e., the peers who are allowed to read the content of the profile according to the privacy policy defined by u.

The criteria considered for the selection of a trusted peer from the set of possible candidate heavily affects the availability of data. Indeed, a first proposal was to select at random the peers on which the data are placed [23] from the set of possible candidate trusted replicas. However, replicating data on different random users' devices is not enough to ensure high data availability because peers participating in the network are heterogeneous in terms of demands and online behavior. To take into account the dynamic behavior of the users of the DOSNs, simple temporal information related to users can be considered by the replication strategies (such as average session length or online-offline correlation [16, 20]). In contrast, the periodic behaviour of users in DOSNs has not been exploited yet in replica selection strategies. However, the availability patterns of the users of the DOSNs are crucial to better design the data replication strategies of the DOSN infrastructure. For example, as shown in [15, 3], users of the OSNs seem to connect to the service with a periodic trends. This suggest the existence of a periodic pattern where each user is connected at similar times each day. We believe that such availability patterns of users can be exploited to achieve accurate predictions about availability of users contents.

For this purpose, we used a linear predictor to predict availability status of the user (online or not) during a certain future time interval, on the basis of his past behaviour. A linear predictor is defined as a linear combination $f(x^t)$ constructed from a set of k terms $x^{t_1}, \cdots, x^{t_k}$ by multiplying each term x^{t_i} with the corresponding weight β_i:

$$f(x^t) = \frac{\beta_1 x^{t_1} + \beta_2 x^{t_2} + \cdots + \beta_k x^{t_k}}{\sum_{j=1}^{k} \beta_j} \tag{1}$$

In particular, weights β_i for $i = 1 \cdots k$ are named *coefficients* and they are used to specify the weights reflecting the importance of each element. For the purposes of data availability, linear predictors are used to calculate the probability that a candidate trusted peer u is online/offline in a given future period t by taking into account the past availability status of the same user u in k different time instants. The coefficient vector can be used in the fitting process to indicate how much each day contributes to the prediction of the availability status. To be able to compare probability measures, we performed a normalization by dividing it for the sum of the weights β, obtaining a normalized value between 0 and 1. Given the vector of probabilities resulting from the application of Equation 1 on each candidate trusted replica, the framework selects the user who has the highest probability of being online.

We describe the algorithm utilized by the primary trusted replica o to select a new trusted replica peer p for a content c **created by the user** u in more details. The peer p must be currently online and be allowed to access c according to the privacy policy defined on c. Furthermore, we consider the probability that p remains online for some time. Initially, we assume that o is authorized to access the content c (optionally, we can consider $o = u$). As specified in Algorithm 1, user o retrieves the user descriptor D_u in order to get the replica table R_u and to have information about the replicas available for the content c (line [2-4]). Then, user o executes an election procedure which selects a new trusted replica for c. User u get the set of online users F having a friendship relation with u (the owner of the content) and then uses the authorization module of the privacy preserving framework to evaluate whether the user $f \in F$ is authorized to read the content c of u (line 5-8).

The privacy policy of c is evaluated on each user f in the set of neighbors by simulating an access request on a user's content c and it may only return permit or deny (line 9). The linear predictor is evaluated only on the users who have obtained a permit authorization decision (line 10) and in such a case, availability status of f (online/offline) during the last k time instants is exploited for prediction. The selection procedure chooses as trusted replica peer, the user f who has the maximum probability to be online, based on the result of the linear predictor. To make availability predictions for t time steps into the future, we iteratively evaluate the linear combination using the k most recent availability samples of each user.

Algorithm 1 User o executes the selection of a replica for the content c of user u.

procedure SELECT
  D_u=getUserDescriptor(u);
  R_u=getReplicaTable(D_u);
  tr=getTrustedReplicas(c);
  $auth$ =getAuthorization(D_n); ▷ init authorization component
  F =getOnlinePeer(D_n);
  $candidates = \emptyset$,$maxVal = 0$;
  for $f \in F$ **do**
    $result = auth.evaluateAccess(c,\text{READ}, f)$;
    if $result$ = PERMIT **then**
      $val = predict(f, x^t) = (\beta_1 x^{t_1} + \beta_2 x^{t_2} + \cdots + \beta_k x^{t_k}) / \sum_{j=1}^{k} \beta_j$
      if $maxVal > val$ **then**
        $candidate = f$;
      end if
    end if
  end for
  $tr(c) = tr(c) \cdot candidate$; ▷ trusted replica selection
end procedure

6 Evaluation

In order to validate our approach in a real scenario, we have implemented a Facebook application, called *SocialCircles!*[5], which exploits the former Facebook API (supported till 1st May 2015) to retrieve different information from the registered users.

6.1 The Dataset

We used a dataset containing the same information described in [9] but collected during a larger period of 32 days. In detail, we sampled all the registered users and their friends every 5 minutes, for 32 days (from 9 March to 10 April 2015). Using this methodology we were able to access the temporal status of about 204 registered users and of their friends (for a total of 44.200 users). A discrete time model is used to represent the availability status (i.e. online/offline) of the users during the simulation. In particular, each day of the monitored period consists of a finite number time slots (i.e., 288 time slots each of 5 minutes), for a total number of about 9200 time slots in the whole monitored period.
In order to apply all our privacy policies, we used a subset of 62 registered users and of their friends (for a total of 23.428 users). In Figure 2 we show temporal information of users by plotting at each time slot the number of online/offline registered users (first plot), the average number of online friends (the second

[5] https://www.facebook.com/SocialCircles-244719909045196/

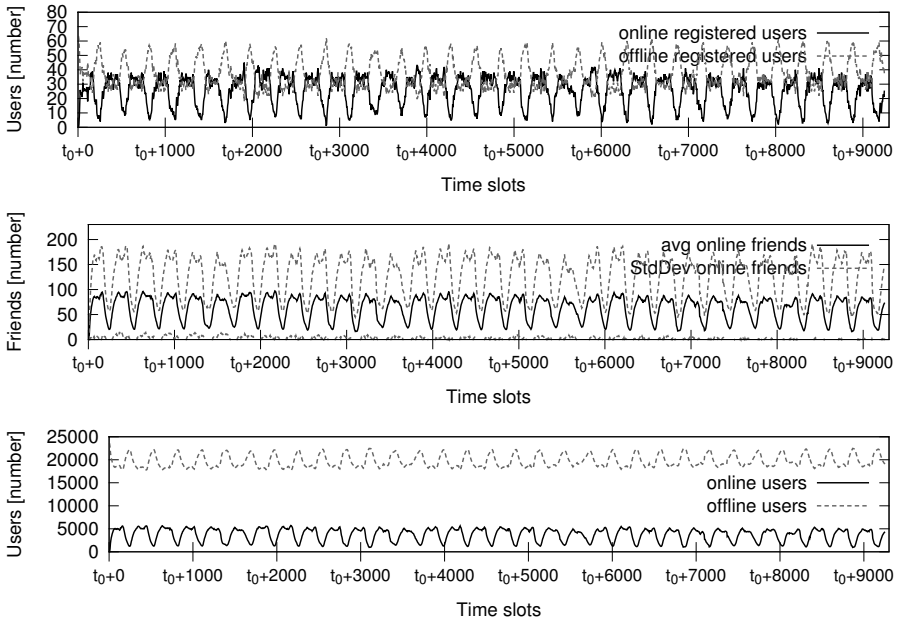


Fig. 2. Number of online/offline registered users, friends of the registered users, and total number of user over the time.

plot), and the total number of online/offline users (the third plot). As shown by the first plot, the number of registered users connected at the same time to the OSN ranges between 5 and 40 and it clearly exhibits a periodic pattern which depends on the time of the day. As we expected, users are more connected to Facebook during the daylight hours, while at most 10 registered users are simultaneously connected during the nighttime hours. The second plot of Figure 2 shows that registered users have an average number of online friends ranging between 20 and 100. In addition, the values of the Standard Deviation (StdDev) indicated a large variation and heterogeneity of our analyzed sample. As showed by the third plot, the overall number of users connected to the OSN ranges between 600 and 5000.

6.2 Experimental methodology

In order to evaluate the proposed approach we used the PeerSim Simulator[6], a P2P simulator written in Java. We developed a set of simulations, based on the dataset previously described, that implement the proposed replication strategy. The duration of the simulation is equal to the length of the monitored period (i.e., 9200 time slots) and the availability status of a peer corresponds to the availability status of a user. A simple DHT-like implementations is used

[6] http://peersim.sourceforge.net/

to bootstrap peers and to track where the contents of a user's profile are stored (i.e., the user descriptor and the replicas table). The number of profiles in the DOSN is constant and equal to the number of registered users, while Posts and Images are generated with equal probability. During the simulation, users publish their contents and select at most k trusted replicas, with k equals to 4, to increase the availability of each content.

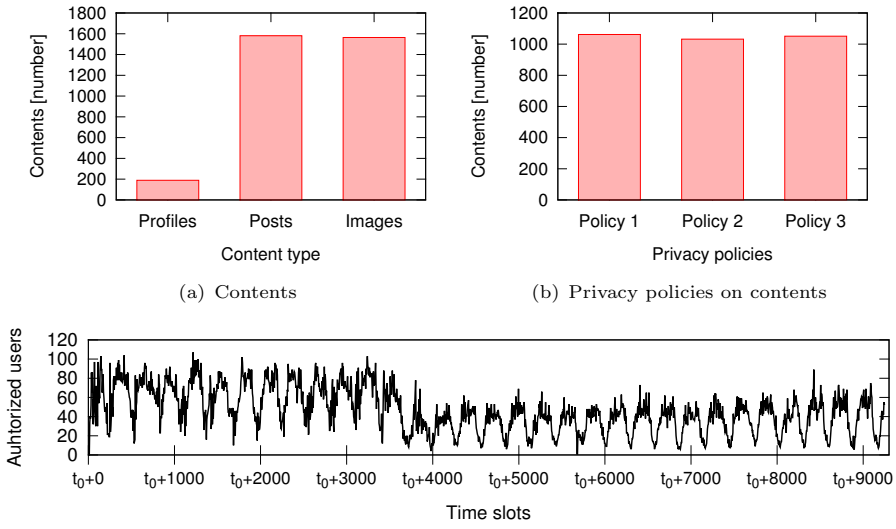
In order to implement the proposed strategy, each content must be linked to a privacy policy which specifies the set of authorized users. For this purpose, we exploit some reference policies which use attributes to model friendships, common friends number, and the strength of the relationship in terms of Dunbar circles, which is a representation of the intensity of the relationship between two users. The strength of the relationship is approximated by using the number of interactions occurred between users [9]. For the sake of clarity, we avoid to show privacy policy by using a proper Privacy Policy Language [10] and we express them in natural language. Consider the user Alice and a content c of her profile. In the experiments, we consider the following reference policies:

Policy 1 Only users who have a friendship relationship with Alice can read c.
Policy 2 Only users who have a friendship relationship with Alice and at least f common friends with Alice and can read c.
Policy 3 Only users who have a friendship relationship with Alice can read c provided that they are in a specific Dunbar circle C.

6.3 Experimental results

From time t_0 to time $t_0 + 3500$ the simulation is initiated and each user registered to our Facebook application creates an empty profile which can be used to publish the shared contents. The future online presence of a user u at a certain time unit t is computed by using a linear predictor, which exploits the availability patterns of user u during the previous 12 days to predict the availability status of u in a future time intervals. For this reason, the first 12 days of the monitoring periods (from time slot t_0 to $t_0 + 3500$) are considered as training set. We introduce also a coefficient vector that specifies the importance of each of the previous 12 days. In our simulation, each past day $j = 1 \dots 12$ has the same importance in predicting the user's availability and each weight β_j of the coefficient vector is equal to 1.

At time $t_0 + 3500$ the set-up phase is finished and users start to publish either Posts or Images with a probability of 0.5. Figure 3(a) shows the total number of Profile objects created during the simulation as well as the total number of Posts and Images published. The number of profiles in the DOSN is constant and equal to the number of registered users (i.e., 62), which amount to 189 data objects. The total number of Posts and Images published on these profiles does not exceed 3.145 contents (1.581 Posts and 1.564 Images, respectively) and each registered user publishes an average number of 25 Posts and 25 Images on its profile. When the user creates a content, it assigns to the generated content a privacy policy randomly chosen among those previously defined. In Policy 2,



(a) Contents

(b) Privacy policies on contents

(c) Average number of authorized users

Fig. 3. Statistics on the contents created by users during the simulations.

the number of common friends (f) is equal to 1. Figure 3(b) shows that the three policies are uniformly distributed among the contents. Figure 3(c) shows the average number of users that can access a content. Initially (from time t_0 until $t_0 + 3500$), each user's profile is empty and it can be accessed by all the user's friends. Thereafter (i.e., from time $t_0 + 3500$), new contents are created and only a subset of the users' friends are enabled to access them, according to the selected privacy policies.

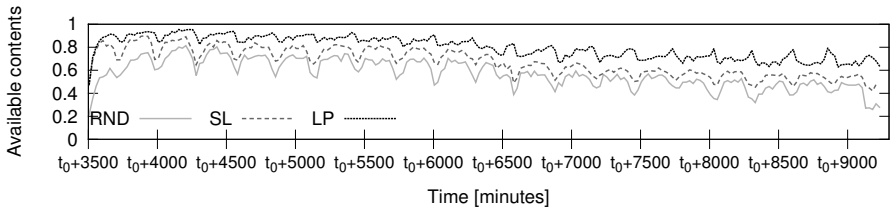
With the previous set up, we investigated the availability resulting from our approach with respect to other peer selection strategies. In particular, during the simulation, users select at most $k = 4$ trusted replicas to increase the availability of each content c. Each trusted peer hosting a replica for c is chosen by using the following strategies:

RND: Randomly selects a trusted replica from the set of user's peers authorized to access c.

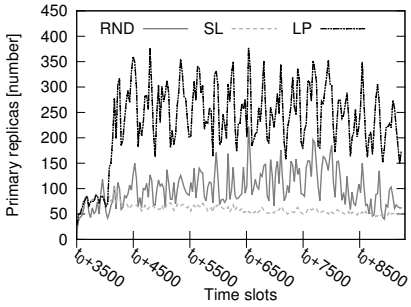
SL: Selects as trusted replica a user's peer that is authorized to access c and that has the highest average session length.

LP: Selects as trusted replica the user's peer authorized to access c and that has the highest probability of remaining online (see Algorithm 1).

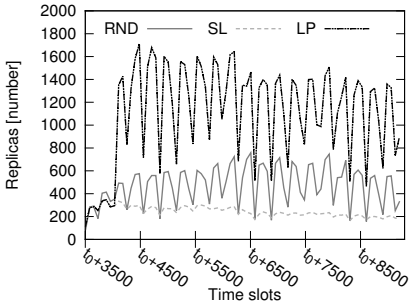
We measured the number of trusted replicas and the availability of contents at each time slot and for each of the proposed strategies. Figure 4(a) shows the percentage of contents available on the DOSN for each time slot of the simulation period. The LP selection strategy outperform over the SL and RND strategies by reaching the availability of the most part of the contents (about 95%) dur-



(a) Availability



(b) Primary trusted replicas

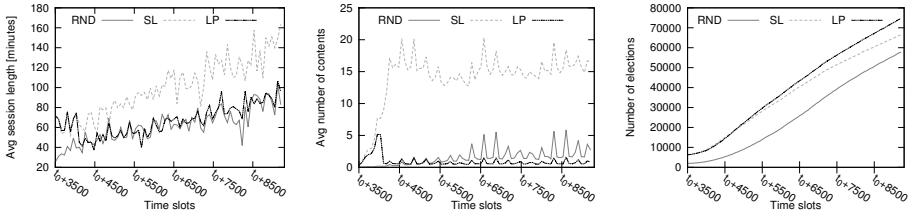


(c) Trusted replicas

Fig. 4. Statistics on the availability of the contents created by users during the simulations.

ing the daylight hours. Clearly, the availability of the contents depends on the number of users connected at the time of the day and it decreases of about 10% during the nighttime hours. The availability of the contents provided by the SL strategy is lower than LP: at most 80% of the contents were kept available. In particular, the plot indicates clearly that during the night periods the LP strategy has a gain of about 30% with respect the SL strategy. Indeed, the LP strategy exploits availability patterns of users to predict the users' peers that will be online (with high probability) during the nighttime hours. In contrast, the SL strategy chooses the user's peer with the highest average session length, regardless of the time of the day in which the user spends this time.

Since the simulation selects at most 4 trusted replicas for each content, the total number of trusted replicas available in the DOSN is bounded by $4 \cdot \#Contents$, where $\#Contents$ represents the total number of contents created by users (i.e. 3334 contents). Figure 4(b) and 4(c) show, respectively, the number of primary trusted replicas and trusted replicas. The LP strategy employs at most 400 primary trusted replicas for providing the users' contents, while trusted replicas created as support for availability of such contents ranges between 600 and 1600. Instead, the total number of primary trusted replicas and trusted replicas of the RND strategy is quite low (about 100 primary trusted replicas and less than 600 trusted replicas) as the most part of the contents are not provided. The SL strategy uses the fewest number of primary trusted replicas and trusted replicas because it always selects as replicas the users' peers with the highest average



(a) Avg session length of the trusted replicas

(b) Contents stored on trusted replica

(c) Number of elections

Fig. 5. Statistics on the number of the messages and the storage load taken by the trusted replicas.

session length. We investigated the average session length of the trusted replicas selected by each allocation strategy. As expected, the Figure 5(a) clearly indicates that trusted replicas selected by the SL strategy have the highest average session length while trusted replicas selected by the RND and LP strategy have very similar average session length. This facts indicate that LP strategy provides higher availability of contents with respect to SL and RND, despite the average session length of the trusted replicas is lower. We investigated the load of replicas by measuring the average number of contents stored on each trusted replicas. The results given by Figure 5(b) indicate that the SL allocation strategy has the highest average load (about 20 contents stored on each trusted replica) because the contents are mainly stored on the same trusted replicas, i.e. those with the highest average session length. Indeed, the LP and RND strategies balance the load uniformly between the different trusted replicas.
Finally, we assessed the number of trusted replica elections. Figure 5(c) shows the sum of the number of the trusted replicas selected by each strategy during the simulation. The LP and the SL allocation strategy take about the same number of trusted replica selection while the RND strategy perform less elections of trusted replica peers.

7 Conclusion and Future works

In this paper we have presented the design and evaluation of allocation strategies for data in DOSNs using the privacy policies defined on content and the availability patterns of users. In particular, the proposed replication strategy selects the peers that will store a replica of the content c of a user u among the set of users authorized to access c and it uses linear predictor, which exploits past availability status (online/offline) of these peers to predict their online status in a future time. We evaluated the proposed strategy by using a real Facebook dataset and by comparing it with the other allocation strategies used by the current DOSNs. The results indicate that the proposed allocation strategy increases data availability of contents of about 30% with respect to ours competitors. In addition, the proposed allocation strategy gives back to users more control over

their contents since they are stored and maintained according to the users' privacy preferences.
We plan to enhance the allocation strategy by adjusting the number of trusted replica for a content as a function of the number of authorized users available in the DOSN. Finally, a further extension is the definition of proper mechanisms to balance of the load on the trusted replicas.

Acknowledgements *This work has been partially funded by the project Big Data, Social Mining and Risk Management (PRA 2016 15), University of Pisa.*

References

1. Aiello, L.M., Ruffo, G.: Lotusnet: Tunable privacy for distributed online social network services. Computer Communications 35(1), 75 – 88 (2012)
2. Bodriagov, O., Buchegger, S.: Encryption for peer-to-peer social networks. In: Security and Privacy in Social Networks, pp. 47–65. Springer (2013)
3. Boutet, A., Kermarrec, A.M., Le Merrer, E., Van Kempen, A.: On the impact of users availability in osns. In: Proceedings of the Fifth Workshop on Social Network Systems. pp. 4:1–4:6. ACM (2012)
4. Buchegger, S., Schiöberg, D., Vu, L.H., Datta, A.: Peerson: P2p social networking: Early experiences and insights. In: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems. pp. 46–52. SNS '09 (2009)
5. Conti, M., De Salve, A., Guidi, B., Pitto, F., Ricci, L.: Trusted dynamic storage for dunbar-based p2p online social networks. In: On the Move to Meaningful Internet Systems: OTM 2014 Conferences - Confederated International Conferences. pp. 400–417. Springer (2014)
6. Cutillo, L.A., Molva, R., Strufe, T.: Safebook: a Privacy Preserving Online Social Network Leveraging on Real-Life Trust. Communication Magazine, IEEE 47(12), 94–101 (2009)
7. Datta, A., Buchegger, S., Vu, L.H., Strufe, T., Rzadca, K.: Decentralized online social networks. In: Handbook of Social Network Technologies and Applications, pp. 349–378. Springer (2010)
8. De Salve, A., Di Pietro, R., Mori, P., Ricci, L.: Logical key hierarchy for groups management in distributed online social network. In: IEEE Symposium on Computers and Communication, ISCC 2016, Messina, Italy, June 27-30, 2016. pp. 710–717 (2016)
9. De Salve, A., Dondio, M., Guidi, B., Ricci, L.: The impact of user's availability on on-line ego networks: a facebook analysis. Computer Communications 73, 211–218 (2016)
10. De Salve, A., Mori, P., Ricci, L.: A privacy-aware framework for decentralized online social networks. In: International Conference on Database and Expert Systems Applications. pp. 479–490. Springer International Publishing (2015)
11. De Salve, A., Mori, P., Ricci, L., Al-Aaridhi, R., Graffi, K.: Privacy-preserving data allocation in decentralized online social networks. In: Distributed Applications and Interoperable Systems. pp. 47–60. Springer (2016)
12. Ellison, N.B., et al.: Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication 13(1), 210–230 (2007)
13. Gao, H., Hu, J., Huang, T., Wang, J., Chen, Y.: Security issues in online social networks. IEEE Internet Computing 15(4), 56–63 (2011)

14. Ghosh, S.: Distributed systems: an algorithmic approach. CRC press (2014)
15. Golder, S.A., Wilkinson, D.M., Huberman, B.A.: Rhythms of social interaction: Messaging within a massive online network. In: Communities and technologies 2007, pp. 41–66. Springer (2007)
16. Gracia-Tinedo, R., Artigas, M.S., Lopez, P.G.: Analysis of data availability in f2f storage systems: When correlations matter. In: 2012 IEEE 12th International Conference on Peer-to-Peer Computing (P2P). pp. 225–236. IEEE (2012)
17. Graffi, K., Gross, C., Stingl, D., Hartung, D., Kovacevic, A., Steinmetz, R.: LifeSocial.KOM: a secure and P2P-based solution for online social networks. In: Consumer Communications and Networking Conference (CCNC), 2011 IEEE. pp. 554–558 (2011)
18. Gross, R., Acquisti, A.: Information revelation and privacy in online social networks. In: Proceedings of the 2005 ACM workshop on Privacy in the electronic society. pp. 71–80. ACM (2005)
19. Guidi, B., Amft, T., Salve, A.D., Graffi, K., Ricci, L.: DiDuSoNet: A P2P architecture for distributed dunbar-based social networks. Peer-to-Peer Networking and Applications 9(6), 1177–1194 (2016)
20. Kermarrec, A.M., Le Merrer, E., Straub, G., Van Kempen, A.: Availability-based methods for distributed storage systems. In: Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium on. pp. 151–160. IEEE (2012)
21. Koll, D., Li, J., Fu, X.: Soup: An online social network by the people, for the people. In: Proceedings of the 15th International Middleware Conference. pp. 193–204. Middleware '14 (2014)
22. Narendula, R., Papaioannou, T.G., Aberer, K.: A decentralized online social network with efficient user-driven replication. In: Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom). pp. 166–175. IEEE (2012)
23. Narendula, R., Papaioannou, T.G., Aberer, K.: Towards the realization of decentralized online social networks: An empirical study. In: 2012 32nd International Conference on Distributed Computing Systems Workshops. pp. 155–162. IEEE (2012)
24. Nilizadeh, S., Jahid, S., Mittal, P., Borisov, N., Kapadia, A.: Cachet: A decentralized architecture for privacy preserving social networking with caching, pp. 337–348 (2012)
25. Schiöberg, D., Schneider, F., Trédan, G., Uhlig, S., Feldmann, A.: Revisiting content availability in distributed online social networks. CoRR abs/1210.1394 (2012)
26. Sharma, R., Datta, A.: Supernova: Super-peers based architecture for decentralized online social networks. In: 2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012). pp. 1–10 (2012)
27. Zhang, C., Sun, J., Zhu, X., Fang, Y.: Privacy and security for online social networks: challenges and opportunities. IEEE Network 24(4), 13–18 (2010)