Online Social Networks and Media 7 (2018) 12–29

Contents lists available at ScienceDirect

Online Social Networks and Media

journal homepage: www.elsevier.com/locate/osnem

Managing social contents in Decentralized Online Social Networks: A survey

Barbara Guidi[a,*], Marco Conti[b], Andrea Passarella[b], Laura Ricci[a]

[a] *Department of Computer Science, University of Pisa, Largo Bruno Pontecorvo, Pisa, Italy*
[b] *Institute for Informatics and Telematics, IIT, CNR, Via Moruzzi, Pisa, Italy*

ARTICLE INFO

Article history:
Received 13 February 2018
Revised 4 July 2018
Accepted 7 July 2018

Keywords:
Decentralized Online Social Networks
Data Management
Data availability
Information diffusion
Privacy

ABSTRACT

The widespread diffusion of Online Social Networks has given unforeseen opportunities for their users to share contents and mutually interact. However, current platforms offer inadequate guarantees as far as concerns the privacy of their users. To address these issues and leave to the users more control on their data, several recent proposals suggest to decentralize the storage of social data, aiming at leaving their control entirely to their owners. This approach has lead to the definition of Decentralized Online Social Networks (DOSNs), ranging from completely decentralized solutions, i.e. P2P solutions, to hybrid systems integrating external and private resources for storing user data. While DOSNs allow users to have more control over their data, they raise new challenges concerning the management and availability of social data. Guaranteeing data availability in a highly dynamic environment, defining proper algorithms for information diffusion and guaranteeing data privacy in a distributed setting are currently open problems in this area.

This survey presents an overview of these challenges and of the main solutions presented in the literature. Existing proposals are classified taking into account the strategies adopted to manage social data, focusing on the data availability and on information diffusion. The survey also presents the new privacy issues arising in DOSNs. Finally, an overview of the main open issues in this research area is presented.

1. Introduction

Online Social Networks (OSNs) are nowadays one of the most popular applications in the Internet. They have attracted a huge amount of users during the last years by changing the way people communicate and interact. Facebook can be considered the most representative OSN with, at the beginning of 2018, more than 2 billion active users, and the highest number of daily users connections. OSNs provide several services [1] offering to their users the opportunity of building a public profile, looking up new friends among the registered users, establishing relationships, and sharing content. Furthermore, these platforms also allow sharing of information within groups of users and the possibility of building communities of users characterized by common interests.

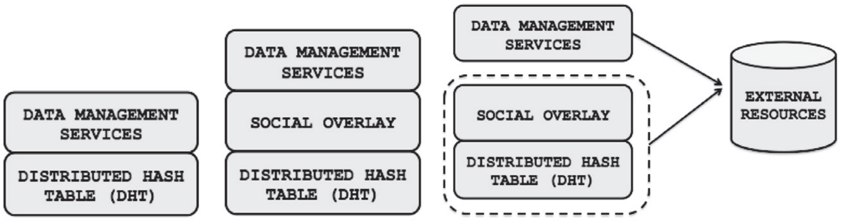
One of the major problems of current OSNs, which are mainly developed on centralized platforms, concerns the privacy of the users' data. Indeed, social data are stored in centralized servers, and the companies running the OSNs, use these data for commercial goals. In May 2015, a report[1] commissioned by the Belgian Data Protection Authority based on the analysis of OSN policies and terms-of-use, concluded that Facebook gives users a false sense of control over their data privacy. More recently, it has been evident to the general public that Facebook data might have been sold without any consent of the legitimate owners. These are only few examples of several legal issues that involve not only Facebook, but also further OSNs, like Twitter, or Google+. Furthermore, centralized OSNs may suffer of other problems, like limited scalability and high maintenance costs to manage data of large number of users [2].

All these issues have led researchers to propose alternative solutions based on the decentralization of OSN services. A Decentralized Online Social Network (DOSN) [3] is an online social network implemented on a distributed platform. In a DOSN, there is no single service provider but a set of nodes that cooperate to guarantee all the functionalities offered by a centralized OSN. Decentralization gives several benefits in terms of privacy. Indeed, there is no central entity that has the control on all users' data or changes the

[*] Corresponding author.
E-mail address: guidi@di.unipi.it (B. Guidi).

[1] https://www.huntonprivacyblog.com/tag/belgium/.

https://doi.org/10.1016/j.osnem.2018.07.001
2468-6964/© 2018 Elsevier B.V. All rights reserved.



(a) DHT-based (b) SO-based (c) External Resources-based

Fig. 1. Distributed Online Social Network architectures.

existing terms of service, and this gives to the users more control over their data.

Moving from a centralized to a distributed architecture gives the opportunity to develop the social network platform by exploiting different distributed models. Following the way of how people interact with each other in a social network makes P2P architectures [4] a natural way to implement DOSNs. However, several alternative solutions are possible, like exploiting network of trusted servers, or mobile and opportunistic networks. Also hybrid solutions where the user exploits both the storage of its own device and a cloud storage service are possible [3].

Researchers have presented several solutions for DOSNs in the last decade. Furthermore, recent years have seen several initiatives to implement and deploy real DOSNs. Just to provide a few examples, the precursor of current DOSNs can be considered Diaspora [5], a solution based on a federation of trusted servers, which has gained a lot of popularity, despite not fully decentralized. Other DOSNs, like Tent and Friendica, are based on similar concepts. On the other hand, Retroshare is a fully decentralized DOSN, designed to provide maximum security and anonymity to its users.

While decentralization gives interesting possibilities for increasing the privacy level of users' data, it introduces many challenges, still to be solved. These mainly concern the management of social data in a distributed environment. With the term "social data", we identify all data exchanged in the Social Networks concerning both information related to the users (contact details, describing the userâs identity, relationships, community memberships, etc..) and generated contents (comments, posts, etc..). One of the main problems is to guarantee the availability of social data, in an environment characterized by a high level of dynamism. Another main problem is related to the development of techniques for propagating social updates in an efficient way. Finally, even if data is no more stored on centralized servers, new privacy issues have to be solved, for instance detecting trusted nodes that may host the profile of off-line users.

This paper reviews the main solutions proposed in the literature to solve these challenges and presents a classification of the existing solutions. We describe in detail the DOSN research challenges, focusing, in particular, on the data management problem. First we introduce a classification of the main architectural solutions for DOSNs. Then, we introduce existing approaches for guaranteeing data availability, and the main techniques used to spread social content among the users of the social network.

Existing surveys or taxonomies [3,6,7] provide a detailed description of DOSNs, mainly from a privacy and security point of view. They address only partially other important issues, such as managing social data in a distributed environment, and do not provide a classification of existing proposals based on this characteristic. On the other hand, we focus on the data management problem in DOSNs and present current techniques for data availability, information diffusion, and data privacy in a distributed environment.

The rest of the paper is organized as follows. In Section 2, after a general overview of DOSNs, we propose a coarse classification of existing systems and we introduce the main research open issues. Section 3 presents the structures used to represent social data. Section 4 introduces current techniques to guarantee data availability and classify them, while Section 5 discusses and classifies the techniques used to spread social information. In Section 6 we describe data privacy in DOSNs. In Section 7, we present relevant examples of DOSNs. Section 8 discusses the limitations of current solutions and open problems and concludes the paper.

2. Decentralized Online Social Networks: overview and key research challenges

A general model for the decentralization of the services in an OSN is given in [8], where a DOSN is defined as a distributed system including a Social Network (SN), a Social Networking Service (SNS) and a Communication and Transport (CT) level. Using a network stack layering representation, SN is located on the top layer, SNS is the intermediate layer, and CT is the bottom one. The SN level provides the common social network functionalities, such as chat, mails, wall posts and/or tweets, etc..., while the SNS is usually implemented by a P2P network, i.e. a distributed network composed of a large number of distributed, heterogeneous, autonomous, and highly dynamic peers sharing their own resources (processing power, storage capacity files contents, etc...). The participants of the P2P network can act as a server and a client at the same time and their functionalities are accessible by other peers directly, without passing through intermediary entities. Finally, the CT level consists of Internet, mobile or opportunistic infrastructures that are used by the above levels to communicate.

DOSNs can be classified by considering the overlay connecting the peers of the users participating to the social network and, possibly, a set of external resources. As shown in Fig. 1, DOSNs can be classified in three categories. A first coarse grained distinction depends on whether external storage services are exploited. Solutions exploiting only the peers of the users participating to the social network can be further classified as:

- solutions exploiting a Distributed Hash Table (*DHT-based system*), whose nodes are those of the users participating to the social network, where the overlay connecting the peers is defined by the specific DHT topology. The DHT can be exploited both to store the social content and as an indexing service.
- proposals where peers are connected through a social overlay (SO) (*SO-based system*), where a logical connection between a pair of nodes corresponds to a friendship relationship. Solutions is this class may also exploit a DHT, generally only as an indexing service.

In those systems that exploit the DHT to store social content, data are stored encrypted in the DHT [9]. Since mapping of data to node storages is guided by an hash function, users can not control where data are stored. Other systems choose "trusted" nodes to store content replica and use the DHT only to index these replica.

In this case, the use of cryptography is not mandatory due to the use of trusted nodes.

Let us define the concept of social overlay in more details. In a social overlay nodes are connected to other nodes corresponding to social acquaintances, as explained in [10]. In the simplest model the links of the social overlay correspond to the links of the social graph, in particular each user is connected to all other users in its ego network. The *ego network* [11] is a social network model typically used to model a social overlay. Let us consider the classical model of OSNs described by a graph $G = (V, E)$, where V represents the set of users and E the set of links modeling the social relations between them. When considering the simplest formalization of OSN, G models relationships self-defined by each user (for instance friendship relations defined by the users in Facebook) and contains an edge between any pair of friends. The ego network of a user represents a structure built around the user that contains its direct friends, known as *alters* and also the direct connections between the alters. Formally, each vertex $u \in V$ can be seen as an *ego* and $EN(u) = (V_u, E_u)$ is the ego network of u where $V_u = \{u\} \cup \{v \in V | (u, v) \in E\}$, and $E_u = \{(a, b) \in E | a = u \vee b = u \vee \{a, b\} \subseteq V_u\}$. $N(u) = V_u - \{u\}$ is the set of adjacent nodes of u.

The simple model where each link of the ego-network corresponds to a link of the social overlay is not realistic, because ego-networks may be very large. In SO-based DOSNs each user defines connections with a subset of the nodes in its ego-network, and the choice of these nodes may be different according to the specific ego-network. In SO-based systems, each user has a set of replicas, but it keeps its data until it is online and transmits them to the requesting friends through the social overlay. When the user is offline, the set of replica nodes are responsible to manage data on behalf of the owner.

Note that, actually, many DOSNs are based on an hybrid architecture that exploits both a social overlay and a DHT for managing the overlay, i.e. for bootstrapping new users, for look-up services, and to guarantee a complete connectivity of the network.

Finally, due to the advent of cloud services, several DOSNs integrate the distributed layer with external resources, such as cloud storage services. In the following, we will refer this second class of solutions as *external resource-based proposals*.

2.1. Research challenges

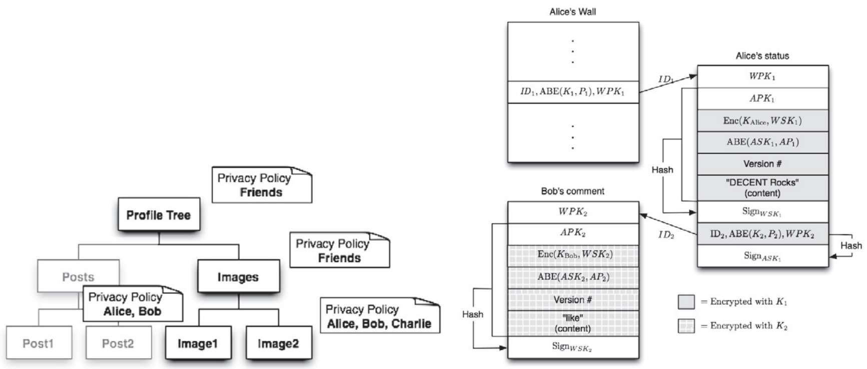
Decentralizing the functionalities of the OSNs requires finding solutions to tackle the issues associated with the decentralization. The focus of this survey is mainly on the challenges associated to the *Data Management Services* for DOSNs. With this term we refer to the set of functionalities needed to guarantee data availability, the diffusion of social information, and the protection of social data. In a centralized OSN, the servers guarantee both the availability and the diffusion of the information, with a cost in terms of privacy. In a distributed system, such as a P2P network, guaranteeing both the data availability and information diffusion is a challenge due to the dynamic and distributed nature of these systems. Specifically, the main challenges for defining the Data Management Services are the following ones.

- *Data availability/persistence*.
  Current social networks allow a user to set its presence information (e.g., being online, busy, offline) and share this information with the other ones. Some social network rely on users to select their presence status, while in other applications, mainly mobile applications, the status is often inferred from the activity of the user and set to offline when the application is executed in background. Instead, in this paper, whenever we hereafter refer to a user being online, we mean that the user's application is running in foreground or in background, and does not refer to the current presence status of the user. Similarly, we refer to a user being offline when its application is not running because the user has logged out.
  The problem of data availability occurs because data in a DOSN are stored on users and are available only when the user is online, while they should be always accessible, even when it is offline. Without a centralized storage, data should be therefore replicated on other nodes to be always available. This is one of the most important problems in every distributed system, but it is particularly important for DOSNs, where social contents are very frequently required by users and the amount of new contents is very high (if we consider OSNs, such as Facebook, the last year there were 500,000 Facebook likes every minute and 1.3 million pieces of content were generated every minute of every day[2]).
  The high level of dynamicity of DOSNs infrastructures makes the problem more challenging. Dynamism is a well-known characteristic of distributed systems, like P2P or mobile networks, where peers can enter and leave the system at any time, even if dynamism of users of a social network has some peculiar characteristics. Note that two types of dynamism can occur, at different levels. The first type of dynamism regards the infrastructure and is related to the churn of users corresponding to the user logging in/off the social application. Therefore, in different snapshots, the social overlay may change due to the users' churn.
  Furthermore, since social relationships can change due to the addition and/or the removal of relations between users and of the number of users of the DOSN, the structure of the social graph changes during time. This second type of dynamism is present also in centralized OSN, but in DOSN it may have an impact on the structure of the underlying overlay, especially for SO-based proposals.
- *Information diffusion*. In a OSN, each update from one user must be notified to all its direct social contacts and beyond (i.e. friends of friends), depending on the type of social network. In a DOSN, the definition of mechanisms for the diffusion of data is therefore fundamental. These are required, for instance, to manage users' updates, i.e. status updates, published posts, comments and so on. Each update from one user must be propagated to all its social connections in a scalable and efficient way.
- *Privacy*. Data can be stored not only on the node of profile owner, but also on others which can be both known and unknown nodes. The classical requirements for privacy should be refined to consider the context of DOSNs. Moreover, in an OSN, users' data may be characterized by different levels of privacy. Without specific mechanisms for guaranteeing privacy, data could be accessed by everyone so causing privacy breaches. Encryption is usually used in DOSNs to keep control over social contents. However, the classical requirements for privacy should be refined in the context of DOSNs.

Other important challenges are summarized in the following list:

- *Scalability*. Mapping a social graph onto a distributed network can be very expensive due to the number of social links for each node, so the cost of mirroring the social network links into distributed network links can be high, and it can be very inefficient due to the high number of inactive links.
- *Topology*. In classical P2P applications, such as file sharing, nodes are connected with unknown nodes in the network and

[2] https://socialpilot.co/blog/125-amazing-social-media-statistics-know-2016/.



(a) Hierarchical organization of social contents.

(b) An example of a profile in DECENT.

Fig. 2. Social contents representation.

they exchange content with any other nodes in the network. In a DOSN, the information about social relationships is important and nodes may be connected according to their social relationships. This should facilitate operations such as information diffusion or data storage, and also the definition of trusted overlay connections. As a downside, given the possibility of a relatively small set of friends, this would limit the availability and robustness of data access. For these reasons, finding an overlay topology that exploits social relationships and, as the same time, overcome previous limitations, is a current challenge.

- *Physical locality*. Today, mobile devices are the most used way to connect to the Internet, and in particular to the OSNs. The introduction of these devices has enabled the creation of mobile social networks connecting people that are physically close to each other. This opens the way to novel challenges, for instance taking advantage of users' close proximity for the definition of ad-hoc virtual communities or detecting the periodic physical locality patterns of the users to improve their knowledge about opportunities to socialize.

3. Social data representation: an overview

In this section, we briefly review how social data is generally represented. Like in OSNs, also in DOSNs a user is represented by a profile that can be considered his/her digital proxy, stored on one or more nodes. In detail, a profile represents the virtual identity of the user, and usually contains personal information, like private information (name, date of birth, town, etc.), friends list, posts (or tweets), comments, videos and photos [12]. DOSNs adopt several solutions to represent users' profiles as containers of social contents.

In [13] a social profile is hierarchically modeled by a tree whose nodes correspond to the social contents. Each node is paired both with a unique identifier and with a privacy policy describing the access policies to that content, while the root of the tree is the entry point for all the data contained in the user's profile. Fig. 2 a shows the hierarchical organization of a user's profile, including, at the first level, the post and images content categories, and, at the second level, the specific posts and images. Each node of the tree is paired with a privacy policy that defines a set of authorized users.

In DECENT [14] a profile is a root object, which contains references to container objects (Fig. 2b). A social content like a status update, a shared link, a photo or video, or a photo album is stored in a container object that has two components: the main content and a list of comments/annotations. Indeed, each profile is hierarchically organized, but without enforcing a tree structure. Indeed a single object can have multiple parents objects (i.e. a photo or a video).

In LifeSocial.KOM [15], social contents, i.e. Data Objects , contain atomic information, like a profile or a photo, and also additional links to other objects, for instance a Data Object representing an album contains links to a set of photos. Distributed linked lists are used to store the objects linked to the original atomic object.

In [16] data are managed as items and the system supports two kinds of items: objects and access policies. Objects are the core of the user's profile. An object is immutable, like Facebook wall posts, comments, etc. Every object has a unique descriptor containing information about the owner, a sequential number, and an expression in the access-policy language proposed by Confidant [16]. Objects consists of meta-data header followed by the content. The owner and the sequential number uniquely identify the object.

4. Data availability

One of the main challenges for distributed systems in general is guaranteeing data availability when the application of the data owner is offline. Several techniques have been proposed to solve the data availability problem in the general context of distributed systems; for instance, distributed caching [17,18], dynamic data replication [19] and erasure coding [20]. However, traditional approaches can not be easily adapted to DOSNs because of the different usage patterns [21]. Indeed, OSN applications online sessions are different with respect to temporal patterns of other applications, for instance file sharing or content distributed networks [22], and for this reason, techniques such as erasure coding are not easy to implement.

Moreover, users' social data are generally accessed only by specific sets of users, for instance its friends or friends-of-friends. These peculiar characteristics of DOSNs may be exploited for managing social data storage and retrieval. For instance, a user could choose its friends for storing its private contents, because these nodes are considered trusted and can also exploit the knowledge of their temporal behaviour to choose the best node(s) where to replicate the data.

In the following sections we describe the techniques currently used to guarantee data availability in existing DOSNs which will be presented in detail in Section 7. Fig. 3 presents the main classification of current proposals that follows the DOSNs categorization shown in Fig. 1. We describe for each category the main characteristics used to implement data availability approaches. For instance, as described in Fig. 3, DHT-based approaches use the DHT as

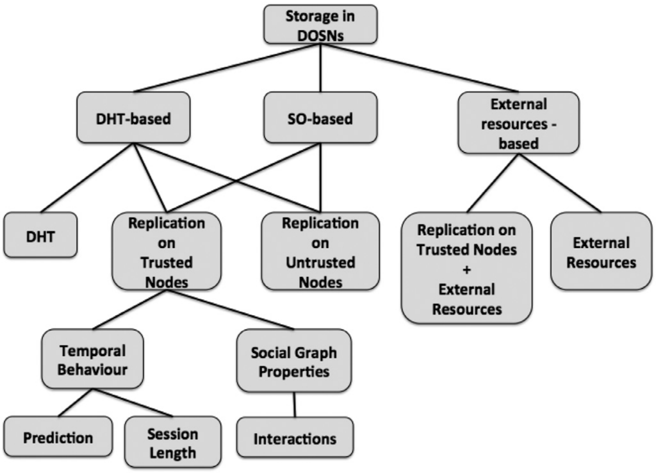

Storage in DOSNs
DHT-based
SO-based
External resources - based
DHT
Replication on Trusted Nodes
Replication on Untrusted Nodes
Replication on Trusted Nodes + External Resources
External Resources
Temporal Behaviour
Social Graph Properties
Prediction
Session Length
Interactions

Fig. 3. Data availabilityin DOSNs.

storage or they exploit it as a globally search-able information directory where they store information about pointers to replica nodes, chosen among both trusted and untrusted nodes. The set of trusted nodes is chosen by using temporal information and social graph properties. The various parameters used in selection replicas can be classified as showed: prediction and session length are used as temporal information and interaction between users is chosen as property of the social graph. Solutions presented hereafter are typically designed to be general and not specific to any type of content, such as photos or profile information.

4.1. Data availability: storing content on the DHT

DOSNs, which use a DHT for storing social data, rely on the underlying DHT replication mechanisms for data availability. The DHT guarantees a high level of availability by distributing and replicating data among the online peers. In this way, data are always available and usually, they are also evenly distributed among the peers, depending on the load balancing mechanisms of the underlying DHT.

However, the use of a DHT for storing social data in DOSNs has several associated problems. Since in a DHT the mapping of data to nodes is governed by the hash function, the user cannot control it and its data are, in general, stored on untrusted peers. This may arise privacy concerns similar to those of a centralized solution, which are generally tackled by exploiting cryptographic solutions.

The second concern regards the huge amount of social data generated by current OSNs which may produce a lot of traffic in the underlying social overlay. For example, Facebook users generate 250 million posts per hour and users like over 4 million posts every minute[3]. With a high level of social contents generated by users, the performance of a DHT are reduced due to the continuous production of new contents. Several "latent" interactions may also be present, [23], which are passive actions such as profile browsing that cannot be observed by traditional measurement techniques. Latent interactions affect the traffic of social contents in the network and they are much more prevalent and frequent than visible events.

When a content is stored on a DHT, each access to social data implies a sequence of routing steps to retrieve it. Though most DHTs define efficient routing algorithms, the traffic on the network increases by a factor proportional to the number of routing hops required to access a content.

Finally, the high dynamicity of DOSNs implies a high maintenance cost for the DHT, in particular a high overhead for maintaining the data structures required for the routing service [24].

4.2. Data availability: exploiting trusted nodes

An alternative solution replicates users' data on trusted nodes. This solution is generally implemented in SO-based DOSNs, where data is generally stored on nodes connected to the user's node through social overlay connections. However, common P2P replication techniques are not suitable in a SO-based DOSNs due to the different characteristics of the online behaviour of social users compared to other P2P applications, such as file sharing. Indeed, the online patterns of DOSN users are more discontinuous than in traditional decentralized applications [22,25]. Considering the use of trusted nodes, the availability of data is strongly dependent on user behaviour. The analysis of the users' behaviour is essential to guarantee a high level of social-data availability [12,26,27], and current proposals exploit the behaviour of users to define specific replica selection strategies. The most important parameter for the replication strategies is the temporal behaviour of users. As explained in [28], availability is not well-modeled by a single-parameter distribution, but instead it is a combination of several time-varying distributions: short-term daily joins and leaves of individual hosts, and long-term host arrivals and departures. This characterization is a consequence of the different behaviour of users in a DOSN. Indeed, as explained in [25], user sessions are shorter than 20 min (median value of 24 min). Only a few users sessions have a long duration, exceeding two hours. In Section 4.4 we describe the selection strategies used to guarantee the data availability when trusted nodes are used, even when the approach exploits also a DHT.

4.3. Availability in External Resource-based Solutions

External Resource-based DOSNs offer a compromise between the use of a centralized server and a completely decentralized system. Data availability in External Resource-based DOSNs is guaranteed by exploiting external resources, generally cloud services. Cloud services are currently a well-known solution because they provide a storage service always available and which can scale up and down to address any storage capacity requirement.

Several solutions exploit hybrid solutions that combine distributed storage with cloud services. One of the main motivations for a hybrid solution is to increase the Quality of Service (QoS). In [29], authors show that a pure distributed storage, like the ones used in SO-based DOSNs, may have poor QoS. To address this, [29] proposes F2BOX, which allows to tune the QoS level by tuning two parameters: the target level of data availability and the fraction of data stored in the Cloud.

In [30], a Cloud-assisted data replication (Cadros) is proposed which uses full replication to increase the data availability and erasure codes to store data in the cloud. In detail, Cadros employs two data replication techniques: all data replicas generated by full replication are stored among friend nodes, while conventional erasure coding techniques are used to store data segments in the cloud, to avoid the complexity of using encryption. By applying the erasure code, the original data are split into m data segments, which are then encoded into n new data segments. When a user needs data, only r data segments of the n encoded segments can be used to reconstruct the original data. In Cadros, nodes can be online or offline, and the online and offline sessions follow certain probability distributions. Moreover, users publish data following certain probability process (e.g., Poisson process). This probabilistic behaviour is taken into account to generate the full replicas, and it is used to predict the values of two metrics: the storage

[3] http://wersm.com/how-much-data-is-generated-every-minute-on-social-media/.

capacity that the friend circle can contribute to and the amount of data friends requests can update at a future time point. When a user publishes content, the data replicas are created and stored in the storage pools, which are either the online friends or the Cloud server. If the storage capacity is unlimited, the newly published data will just be added. Instead, if the storage capacity is limited and/or the storage space is already full, the oldest data will be replaced with the new one.

In [31] three schemes for storing data are proposed: a cloud-based scheme, a desktop-based scheme and, a hybrid scheme. The three schemes differ in the placement of Virtual Individual Servers (VISs) and personal data. In cloud-based decentralization, VISs are hosted by an infrastructure such as Amazon Elastic Compute Cloud (EC2), and the main advantage of this scheme is its high availability. Unfortunately, hosting a VIS in a cloud is currently costly. In the second approach, VISs run on desktop-class machines owned by users. This solution is partially distributed and has a lower cost, but it suffers from lower availability due to the high churn of users. To increase availability, authors propose the third approach which is obtained by combining the desktop-based scheme with a cloud-based service. This approach has the potential of combining low cost and high availability.

4.4. Replica selection strategies

As said before, replication is the most used technique to manage the data availability problem in DOSNs. A replication technique is characterized by a selection strategy used to choose where to replicate the data (replica nodes).

In this section we do not discuss replication strategies for DHT-based DOSNs which store content only on the DHT, because, in this case, replication strategies are those of the underlying DHT and are not under the control of the DOSN. However, we consider all the strategies that involve trusted nodes, even when the DHT is used.

In recent years, several replication strategies have been proposed which may exploit different policies for replica placement: [7]:

- strategies based on trust relationships of a user with other ones.
- strategies based on specific parameters. The most relevant is some statistic measure of the online sessions of the friends' application.
- replicas chosen by the user itself.

The two major users' features used to define a replication strategy are the following ones [12]:

- Temporal behavior. The analysis of the behavior of the users in terms of online presence in a system is defined as *temporal behaviour*. In a social network, both centralized and distributed, this concerns the user's usage of the system, which may be active, by producing social contents, or only passive, when the user only access the social content of the other ones.
- Social graph. The social graph may guide the replication strategies, by exploiting social information such as friendship type, common interests, closeness, trustworthiness, tie strength, etc. The two main features generally exploited to define a replication strategy are: the friendship relationships [32–36] and the trustworthiness, usually estimated by the level of social interactions [12,13].

Before investigating the different replication strategies more deeply, we introduce the main metrics used to evaluate the effectiveness of a replication strategy in DOSNs: *Pure availability* and *Friend Availability* [33]. These metrics are proposed because in a DOSN the availability can be considered not only as the availability of data for all the users, but also as the availability of the data for a set of interested users, called friends. Pure Availability measures the fraction of time data is available in a given time period. In the context of DOSN, however, people interested in the user's data are primarily its friends. For this reason the Friend Availability has been introduced to measure the fraction of time a user's data is available when its friends are online.

Other metrics concern the replication degree and the peer load [32]. The replication degree is the number of replicas hosting a userâs profile, instead the peer load measures the storage load of a peer due to the number of social contents assigned to it.

4.4.1. Replica selection guided by temporal behaviour

The main difference between all strategies in this category is how the temporal behaviour is modeled.

A common problem in designing the replication strategies based on temporal behaviors is the lack of public available statistics about the online behaviour of users in existing OSNs. However, in the last few years, some studies have been conducted. In [37], an empirical study of the various system properties of DOSNs and the parameters that influence them have been investigated. Authors explained that one of the most important parameters to characterize data availability is the online time of a user, while the properties of the social graph are generally used to minimize the number of replicas, or to manage the load balancing.

A typical information used to define a selection strategy exploits the session length, as presented in [12]. Several studies generate synthetic models of the temporal behavior of a user, and of the corresponding sessions length, because of the lack of real dataset. These approaches usually take into account the behavior of the users in other distributed applications. For instance, in [37] three different user session time models are presented. Authors assume that a user could be:

- *Sporadic*: a user is online several time a day sporadically. Authors adopt a fixed session length of 20 minutes, as a conservative choice, on the basis of the results collected from the Orkut social network.
- *Continuous-Fixed Length*, all users are assumed to be online, each day of the week, during a continuous time window of a fixed length; the time slot of a user session is centered around the majority of the observed time of their activity.
- *Continuous-Random Length*, it is the same of the previous model, but each user randomly chooses its own online session length time window in the range [2,8] hours.

The *Sporadic* model is considered by the authors as the most realistic one. In [33], the session length is modeled, independently from the session starting time, using the Weibull distribution.

Some approaches exploit evaluations based on real datasets. For instance, [25], analyses a Facebook dataset containing the online user behaviour. Real data show a daily temporal pattern of users and each user has, on average, less than 100 daily sessions, while the average number of daily sessions for user is less than 4 sessions. Almost half of the sessions are shorter than 20 min and a significant percentage (about 34%) of sessions are less than 10 min. Almost 50% of users present an inter-arrival time shorter than 1 hour. These data are confirmed considering a larger period of 1 month [26].

Another current trend is using predictors to foresee the availability of users. A predictor takes into account the history of the online behaviour of a user to predict if a user could be online in a specific interval of time. In [26], a linear predictor is used to choose the replica nodes. In [38] authors define five different features, which differ depending on the set of users considered (individual or global) and by periodicity (daily, weekly, and flat) and combine them with logistic regression, to predict the probability for each individual user to be online in a future time instant.

An example of a replica selection strategy exploiting previous techniques is given in [32]. In this work the users' online time periods represents, with a predefined granularity (e.g., minutes, hours, days), the time period during which they are active on the social network. This parameter can be either a user input or it may be approximated from the users activity history by exploiting one of the techniques previously discussed. The replica selection algorithm chooses the minimum number of replica, chosen among trusted friends (see next section), that maximize the availability of the user profile, on the basis of their online period. The replica selection is modeled as a *set cover problem*, and, since this problem is NP-hard, an approximated greedy algorithm is defined which selects, at each step, the longest uncovered online period.

4.4.2. *Replica selection strategies guided by the social relationships*

Another common trend for the replica selection is the definition of a set of trusted nodes, which are chosen by exploiting friendship relationships (friend nodes or friend-of-friend nodes). Trusted nodes are used as replica nodes. Some proposals consider all friends of a user to be trusted for hosting the users profile replica, while others exploit other notions of trustworthiness. In the first case, since replicating data at all friends could affect the cost of storing data in multiple copies, in particular for users with a large number of friends [27], some further criteria is needed. Most of the proposed approaches delegate to the user the choice of trusted nodes. Only few approaches analyse the interactions between users to define trusted nodes.

In the approach proposed by [32], each user elects a set of trusted users, among its friends, able to store its profile content and also to enforce access control on the access requests to the profile. A subset of the trusted nodes is the *Trusted Proxy Set (TPS)*, which hosts the replica of its social content. Four algorithms are proposed for the computation of the TPS [32], respectively, minimizing the number of replicas, minimizing the update propagation delay, minimizing the access cost incurred in accessing a user's profile and, maximizing the replication gain. Moreover, authors define a set of selection policies [37] which take into account the online behaviour of users and they exploit various criteria, such as the choice of random friends or the choice of the most active friends.

A similar approach is proposed in [12], where trustworthiness and number of replicas leads selection strategies. In a trusted network built by exploiting the interactions between users, authors propose selection strategies to choose trusted Point of Storages (PoSs). This work exploits both social and structural properties of the social graph, but also the users behaviour. Indeed, a numerical value, called *SocialScore* is computed for each user to measure the users suitability as PoS of a specific node, and it takes into account the tie strength, the gain in terms of trusted connections obtained by the election of a node, and the average duration of a user session that it is a easy way to represents the online behaviour of a user.

4.4.3. *Other approaches*

In SuperNova [39], a new node joining the DOSN, initially relies on a set of super-peers to replicate its data. The list of super-peers is public and super-peers are selected on the basis of the list of services they provide, as how much data they can store, for how long they can store (e.g., for one month), the content they want to store, etc. Instead, Storekeepers are lists of users who have agreed to keep replicated data of another users so that when a particular node *n* is offline, the friends of *n* can contact storekeepers to access its data. Hence, Super-peers or Storekeepers, act as bridges between the user and its friends. Super-peers collect information on nodes and suggest to each node that other node can act as a storekeeper for it.

5. Information diffusion

Nowadays, OSNs produce a huge amount of contents and interactions. The strategy adopted to propagate social data plays a key role to deliver data to targeted users with a limited latency, and avoid message duplication. While in legacy OSNs information diffusion can be carried out by the central provider by exploiting the whole social graph, it is still an open research challenge for DOSNs.

The problem of data dissemination in distributed systems like sensor networks, P2P platforms, and mobile networks, has already been investigated. Existing techniques include publish/subscribe methods [40,41], gossip/epidemic protocols [42,43], probabilistic routing based on prediction [44], distributed caching [45], broadcast approaches [46] and Content Distribution Networks (CDN) [47].

Information dissemination in structured P2P networks may exploit publish/subscribe (pub/sub) techniques which rely on rendezvous nodes to bring information from a publisher to its subscribers [48]. In general, subscribers register their interests in a topic and then asynchronously receive events matching their interest. However, rendezvous nodes can become bottlenecks for the application performance and pub/sub methods in structured distributed systems are sensitive to the dynamics of the network because of the need to maintain subscription information in presence of rendezvous nodes churn.

A gossip-based protocol is based on a repeated probabilistic mutual exchange of information between nodes. In general, gossip protocols are the right choice when scalability and reliability in spite of high level of failures are major concerns. Gossip protocols are more generally adopted in unstructured systems, because they are less susceptible to user's churn and more robust to failures and ensure, with a high probability, that a node interested in a content will eventually get it. Furthermore, they allow to tune several parameters according to the required reliability level.

Hybrid solutions exploiting gossip in structured networks, mainly for spreading information, are also possible.

In the following sections, we will restrict our analysis to the information diffusion techniques currently used in DOSNs to disseminate social content updates.

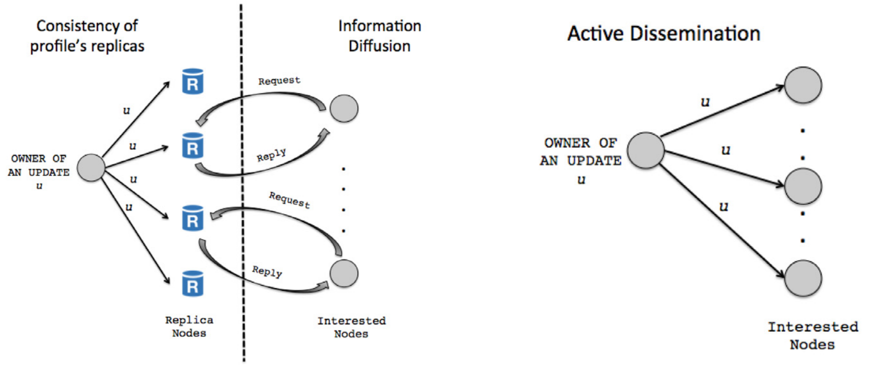
5.1. *Information diffusion in DOSNs*

In a DOSN, the information diffusion is related to the *update dissemination process*. Users' profiles are continually updated and these updates must be distributed in the network to make it available to all the interested users.

Information diffusion methods in DOSNs may be classified as *Request-Reply* or *active dissemination* approaches. In a *Request-Reply* method, an interested user explicitly sends a request to obtain a profile update. This solution is generally exploited to ask for social contents to nodes storing replica of the users' profiles. The node responsible for the content replies to the request providing the update to the interested node. An example is shown in Fig. 4a. When the owner creates an update, it sends it to the replica that stores the update. The interested nodes may ask for that content to one of the replicas, when the owner is offline.

Instead, an *active dissemination* method requires each content to be actively propagated by the node that has generated or updated it. Content updates (i.e. the news feed of Facebook), are sent by the node generating the update to the others, by considering the structure of the social graph and potentially the privacy policies. These methods generally exploit epidemic protocols that have been proved to be reliable and robust for data dissemination in P2P and wireless networks, or broadcast dissemination protocols (Fig. 4b).

A classification of the main methods for information diffusion in DOSNs, is shown in Fig. 5. The classification takes into account



(a) Request-Reply methods with replica nodes.

(b) Active Dissemination methods (Broadcast or Gossip)

Fig. 4. Information diffusion methods in DOSNs.

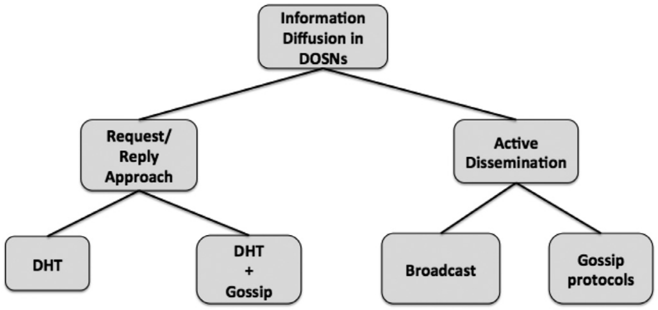


Fig. 5. Information diffusion approaches in DOSNs.

the way replicas are updated in the case of Request/Reply approach and the way information is diffused in the Active Dissemination approach. In detail, the request/reply approach considers the existence of replicas of the information that needs to be updated. The update of replicas is done by exploiting the DHT and by using epidemic approaches, such as gossip techniques. In the Active Dissemination, the information is directly diffused through the interested nodes. The diffusion is implemented by considering broadcast or gossip protocols.

Solutions using external resources combined with distributed storage may exploit one of the previous approaches or a combination of them. For instance, in [49], a cloud-assisted dissemination approach is proposed. When a user generates a content update, this is first encrypted and sent in the cloud, in a space controlled by that user, afterwards, the update is actively disseminated among the nodes via trusted links.

5.1.1. Request-Reply methods

Request-Reply methods are mainly used in systems that use data replication on trusted users, or exploit caching of social data.

[32] shows how replicating profiles guarantees availability. The profile updates are propagated as follows. The replica of a user's profile accepts updates from the user's friends asynchronously and when an update is received, each replica pushes it to to all other replicas. When a replica comes online, it announces itself to all other online replicas and pulls each update that has been collected by the other ones. Eventual consistency is guaranteed by the use of vector clocks. The mapping from a user to its replicas is stored in the DHT, by exploiting as key the unique identifier of the user. When a user u wants to access the profile of a friend f, firstly u accesses the DHT to obtain a reference to the replica of the profile of f, then requests the content directly to a replica that performs appropriate authorization.

In [50] and [51], a Social Caching approach is presented. The authors observe that an active dissemination approach requires to establish $O(n^2)$ network connections (where n is the number of friends), to spread a social update. Therefore, they propose Social Butterfly [50], a system that selects nodes acting as Social Caches to reduce the total number of connections necessary for data dissemination. Social cache nodes act as ”local servers”, while the remaining nodes are assigned as members of one or more social caches to form Social Clusters. A member node only contacts the social caches it is associated with, while a social cache can be a member of multiple social clusters. Content producers push social updates to the social caches they are connected to, while consumers fetch data of their friends from their social caches. This solution is not fully distributed because it requires a knowledge of the entire social graph. In [51] authors propose SocialCDN, an evolution of the previous solution, which is a fully distributed solution that does not require global knowledge about the entire social graph. The problem of minimizing the number of social caches is close to the Neighbour-Dominating problem and the authors propose a fully distributed algorithms to solve this problem. The selection of social caches is social-relationship driven and social caches are selected to cache updates only for friends to ensure security requirements and only allow one hop communication. A similar approach is also used in Supernova.

Request-reply approaches are used also for external resource-based and hybrid solutions. In [16], replica are stored on storage servers, which are selected among the user's friends, while name servers managing the addresses of online replicas and information on replica synchronization are stored on external resources, i.e. cloud services. To store a new data, a client connects to one of the first available storage servers and submits a store-request message. Instead, to retrieve new objects, a client requests the list of online replicas from the cloud server and contact one of them.

In Vegas [52], friend nodes exchange one or more addresses of external services. In case two friends want to exchange a message, one of them selects an external resource and sends the message to it. These external resources act as mailboxes and they can be implemented by email, SMS, instant messaging, and micro blogging (like Twitter).

5.1.2. Active dissemination methods

Several active dissemination methods are based on gossip techniques. The gossip-based update dissemination protocol proposed in [10] uses two gossip protocols to propagate an update: rumor mongering protocol and an anti-entropy protocol. The rumor mongering protocol, is based on a push exchange strategy, where nodes periodically select a neighbour node, and send (push) the content to this node. This approach is used to disseminate updates quickly, however it may generate several messages. For this reason, the protocol is complemented by an anti-entropy push-pull protocol [53]. The protocol runs in background with respect to rumor mongering and guarantees that all nodes that become and remain online for a long enough period eventually receive all updates.

[10] focuses only on the rumor morgering protocol and observes that the standard protocol has to be modified to take into consideration the characteristics of social networks. As a matter of fact, in complex networks clusters are not uniformly distributed like in others complex networks and, for this reason, dissemination proceeds very quickly in the regions characterized by an higher level of clustering and generates lots of duplicates. [10] improves the protocol by piggybacking histories on each message and by defining a set of selection strategies that take into account the topological properties of social networks. Histories are useful in highly clusterized regions of the network, because they record who received a given update and allow nodes to not re-send an update to nodes known to have it. Furthermore, three strategies are considered to select the node receiving the update: a random, an anti-centrality and a fragmentation-aware heuristics. In the first heuristics, a node is chosen uniformly at random as in [53]. The anti-centrality selection strategy assigns to nodes a selection probability that is inversely proportional to their degree so avoiding to favour the propagation in high density regions at the expense of low density ones. The fragmentation-aware heuristics computes the fragmentation of a node as the number of connected components that remain in the graph built by considering the neighbourhood of the node if it is removed. The protocol must guarantee that at least a node for each connected component in the fragmented graph is selected.

In [54] a new epidemic protocol able to spread social updates in SO-based DOSN overlays is proposed. Social updates are spread by exploiting a novel centrality index, the Weighted Ego Betweenness Centrality (WEBC), which discriminates the shortest paths crossing an ego according to the weights of the edges on those paths. The information diffusion algorithm first discovers the community in the 2-hops ego-network of a node. When the node generates a social update, it selects a neighbour for each community and sends a copy of the update to it. The diffusion of the update inside the community is performed by a gossip protocol guided by the knowledge of the WEBC of its nodes computed with respect to nodes that generate the social update.

GoDisco [55] focuses on dissemination of social data by both using exclusively social links and exploiting a semantic context. Nodes inform their neighbours about their interests and they keep track of the behavior of their neighbors. This knowledge is used in the dissemination phase by mapping the semantic value of a message to the interests of the neighbour nodes. As an extension of this work, in [56] authors proposes GoDisco++. The novelty is the exploration of a multi-dimensional social network, whose semantics can be exploited to achieve better dissemination characteristics. Social triads are exploited to avoid duplication among nodes. The essential idea is to avoid forwarding a message to the common neighbors of the node. Furthermore, the approach uses a feedback derived locally by nodes by exploiting the experience gained from previous disseminations.

In [57] an approach exploiting selective propagation of social data is proposed. It takes into account interactions between users to model the strength of relationships in an area of specified interest between interacting users. Users tag their updates according to the interest area. For the propagation of social data belonging to a specific interest, the strength of relationship between users is computed and it is increased or decreased according to the interaction between users. Initially, the system has an equal value of relationship strength for all the users, so that propagation of social data reaches all friends of the user. Afterwards, all changes to the relationship cause a selective propagation of data, according to an interest level and filtering of irrelevant data.

5.1.3. Hybrid approaches

Some approaches exploit both request-reply and a gossip protocol to acquire knowledge of social data. In Cachet [58] data are stored encrypted on a DHT. However, since access to contents stored on the DHT can take hundreds of seconds, and thus it is unpractical for a real social network, a gossip-based social caching algorithm is also exploited. In this way, social trust relationships are exploited in combination with the DHT, to optimize the performance of the system. When a node joins the system, it performs a set of DHT-look-ups to identify some on-line contacts, then it contacts them to obtain information about the presence status of further mutual contacts. After this process, if the presence status of some other contact is unknown, the DHT is accessed again for retrieving the status of these contacts. New updates are immediately propagated to online social contacts. When an offline user comes back online, a presence protocol is used to locate online contacts and query them directly for updates. Social contents and updates are retrieved by using a pull-push gossip protocol. When a node n is contacted by another node m, it pushes all the cached objects that m is authorized to read.

6. Privacy and Security

DOSNs effectively address the main privacy concern of centralized OSNs, because personal data are no more under the control of a centralized provider, such as in Facebook, but are managed by the users themselves. Users do not depend on an external OSN service provider to maintain their data, and the absence of a single control point cuts out several privacy breaches. However, the decentralization of OSN services raises new privacy and security challenges that were previously addressed by the central authority.

In this section, we aim at identifying the security problems arising from the decentralization of data. For this purpose, we refine the classification presented in [59] of security and privacy issues in OSNs. We focus our attention on security and privacy issues that concern DOSNs and organize them into the following categories:

- Privacy breaches: all the attacks that try to strike the users' privacy exploiting the huge amount of users' data available in the system.
- Impersonation attacks: the attacks related to the creation of fake profiles used to access data that are not public.
- Viral Marketing: the spamming and phishing attacks that use the OSN services to disseminate unsolicited messages or malicious software to get users' confidential informations.

6.1. Privacy breaches

OSN users produce a huge amount of contents and personal information that they would like to share only with their direct friends. However, in centralized OSN all data are stored in the server(s) of the service provider and users must trust that it protects all their personal information. [59] highlights several privacy breaches: centralized, distributed, and from third party. Here, we focus only on the distributed privacy breaches due to the decentralized nature of DOSNs. To this end, firstly, let's define the security properties that a DOSN should ensure to their users:

Confidentiality. This is the primary security requirement in DOSN. User's data must be protected from unauthorized access

Table 1
Privacy violations through contents' metadata.

Category	Metadata	Possible inferences	Countermeasures
Stored Content	Size	Content type, content property, content fingerprint	Padding, uniform block sizes
	Structure	List-sizes (num. comments, num. pictures)	Clean encryption headers, two-layers authorization, placeholder entries
	Modification History	Frequencies of status updates, commenting, etc.	Noise (dummy change operations)
Access Control Mechanisms	Encryption Header	Content audience size or even identities, num. of friends	Adapted encryption schemes (attribute-based encryption, broadcast encryption)
	Key Distribution	Friend status changes	
	Key reutilization	Same content audience	
Communication Flows	Direct Connections	IP address, usage patterns	Communication anonymisation, caching, noise, careful protocol design
	Structure	List-sizes (num. comments, num. pictures)	Clean encryption headers, two-layers authorization, placeholder entries
	Modification History	frequencies of status updates, commenting, etc.	noise (dummy change operations)

and only those users who are explicitly authorized by the content owner can read it.

Anonymity. The identity of users and the type of relationship between them should not be inferred by observing the events occurring in the DOSNs. Hence, users' identities must be anonymized and remain hidden from third untrusted parties, as well as the actions performed by the users.

Integrity. Users' identity and their data must be protected against unauthorized modification and tampering. DOSNs must ensure that the contents posted by users' friends are not corrupted.

Authentication. The authentication has to assure the existence of real persons behind a registered OSN profile.

To guarantee confidentiality, users need mechanisms to define who can access their data by using access control policies, at different levels of granularity, provided by the system. Only users who have the permission of the owner can read and/or write the contents. Moreover, communication between trusted parties and the information about all users and their actions should remain hidden from any other part, internal or external to the system. The sender and the receiver of the communication must be authenticated and the communication itself must provide confidentiality and integrity.

Currently, most DOSNs address the privacy issues coupling a distributed approach with encryption techniques. This is largely adopted in DHT-based DOSNs, where data is stored on untrusted nodes. Typically, social contents in OSNs are small sized and each data should be encrypted separately for different sets of recipients to achieve fine-grained access control. The decryption of the social contents generated by friends might be quite time consuming, when the number of contents to access is large. As a consequence, the design of current DOSNs may suffer from performance issues arising due to the fragmented structure of the data.

Since asymmetric cryptography is more computationally intensive than symmetric cryptography, encryption systems based solely on asymmetric cryptography are not generally used. Most of the current DOSNs are based on the conjunction of symmetric and asymmetric cryptography, which is much more efficient since the object itself is encrypted using a symmetric cipher, and then the symmetric key is encrypted multiple times with the public keys of the receivers. However, if the number of users accessing the content is high, the overhead required by first encrypting the content with a symmetric key and then by encrypting the symmetric key with the asymmetric keys, is quite significant [60]. The authors in [60] analyzed the existing security mechanisms for DOSNs by comparing them in terms of efficiency, functionality and privacy.

A large collection of papers seek to propose cryptographic mechanisms more suitable for DOSNs. Gunnar Kreitz et al. [61] focus on the problem of cryptographic primitives that hide the users' data but reveal access policies and they introduce Predicate Encryption (PE) [62], like the Attribute Based Encryption (ABE). A user profile is defined as a set of multiple objects encrypted for different users. Additionally, a Bloom filter is used to store users who can decipher the objects. Datta et al. [63] use a threshold-based scheme to address the problem of back up and recovery of the user's private key in a network of untrusted servers. To improve security of the secret sharing protocol they propose a mechanism to select the most trustworthy delegates based on the social relationships among users.

Encrypting the content is not enough to hide all sensitive information from attackers, because they could infer sensitive personal information from the properties of the contents (such as size, structure, packet header, etc.) and, in general, from metadata. Note that, in a centralised OSN, the users content is managed by a single party that acts as an intermediary hiding metadata information from requesting users. Thus, meta-data can be inspected by the central provider, but information is hidden from everyone else. Istead, in a DOSN, several parties are involved in storing and communicating social content, and even if encryption is employed, more parties can access such information and this aggravates the problem.

The authors in [64] identify the privacy problems that arise when metadata of the contents are shown (see Table 1). They classify the problems raised by inference from the metadata into three categories: (i) inferences from the stored content can reveal size, structure and modification history of the related content; (ii) inferences from access control mechanisms might allow conclusions about a user's social events; (iii) inferences from metadata's communication flows (such as direct connection between users or requests for content sharing) allow to infer information about usage patterns, interests or IP address of the users. Finally, the authors list the countermeasures in order to mitigate the described metadata attacks. However, there is not a comprehensive solution that covers all the problems highlighted.

Authentication has to assure the existence of real persons behind registered OSN members. The most important impersonation attacks which can be made in a OSN will be presented in the next section. Most of them can be faced only through the deployment of strong authentication techniques.

Only some DOSN face the problem of defining an authentication service and most of these proposal use centralized identification services. For instance Safebook [8] uses a centralized trusted third party service, the Trusted Identification Service (TIS). A new user has to be invited by one of its real life friends who is already a Safebook member. Then, the user contacts the TIS and provides its identity, together with a proof of owning it. The TIS sends to the user an identifier and a certificate associating the identifier to the user's public key. Even if the use of a centralized trusted third party service seems to be at odds with the idea of decentralization and may be not acceptable from the users of a DOSN, a central authority (trusted third party) seems hard to avoid, at least in the bootstrapping phase. Other mechanisms, like authentication by in-person contacts, when friends meet in real life and exchange keys over their phones can also be exploited. An alternative proposal is that of [65], which exploits OpenID, an open-standard for decentralized ID management. In OpenID, each user is provided with an identity token asserting the user identity. The token specifies the issuing authority and it is digitally signed and may be optionally be encrypted for confidentiality. Unlike system exploiting a single centralized authentication authority, OpenID allows the use of a set of distributed authorities and, therefore, it is a solution more suited for DOSNs.

As concerns anonymity, almost all current DOSNs proposals don't provide any strategies to guarantee this property. An exception is SafeBook, which presents a decentralized approach which faces the anonymity problem by guaranteeing anonymous communications through the Matryoshka overlay (see Section 7.2). Indeed, nodes are connected through radial paths on which messages flow from the outermost shell to the innermost one by obfuscating the communication. Another example of system that provide anonymity is Vegas. Vegas guarantees that a user cannot be linked to some referenced content by increasing his anonymity. Indeed, only the user and his friends know the address of datastores, used to store data (see Section 7.3), and the directory structure to content mapping.

As concerns the integrity property, it should guarantee the prevention of unauthorized data modification. A solution could be used classical cryptographic techniques, such as digital signature, to guarantee integrity of social contents. For instance, in SafeBook, the TIS, explained before, is also used to guarantee data integrity. Indeed, the TIS sends a pseudonym to each user. The pseudonym is used to guarantee integrity and confidentiality to all messages, which are signed using the senders pseudonym private key and encrypted using the receivers pseudonym public key.

6.2. Impersonation attacks

Impersonation threats in OSNs such as creating fake profiles used by OSN hackers to launch cloning attacks, are considered one of the hot topics for OSN security. The intruder seeks to create fake profiles to setup fake relations with the victim's friends and to mount malicious activities such as misusing the reputations of the OSN users. Impersonating user's identity in OSNs is considered one of the open issues from security and privacy point of view considering that these kind of attacks enable the attacker to have access to a huge amount of social data (name, gender, job, education, photos, etc...). The default options in OSNs such as Facebook is that everyone can access to a profile of another user even if does not belong to its friend list.

We list the major impersonation attacks in OSNs:

Identity Theft:a malicious user acquires the credentials of authorized users and acts on their behalf, then the attacker sends requests to the friends of cloned victim. The hope of the intruder is to gain the trust of these users. By establishing friendship relationships with the contacts of the victim the attacker can access the sensitive personal information of these contacts.

Profile Clone: a malicious user builds a copy of another user's profile.

Profile Porting: a malicious user creates a profile under the victim's identity in an OSN where the victim in not present.

Profile Harvesting: a malicious user gathers data on the participant for purposes that the victims have not considered.

Defamation and Ballot Stuffing: a malicious user forges the reputation of a person using the system.

The impersonation attacks are similar in a centralized and in a distributed environment. Currently, no definitive defense can protect against such attacks. Most of the existing proposals suggest to increase users' alertness concerning their acceptance of friend requests, while others exploit challenge-response protocols to differentiate people from bots.

6.3. Viral Marketing and Spammers

Viral marketing [66] is a technique, which has recently achieved huge commercial success, to spread commercial advertisements through social networks links. OSNs are a suitable target for viral marketing because it is possible to reach a large number of users by spreading information through friends and friends-of-friends relationships. Specifically, spamming refers to unsolicited, unwanted (social) contents sent to all the users (*broadcast spamming*) or to a specific subset of them (*contex-aware spamming*). Spammers can exploit social information and services of the social platform to enhance the propagation of their messages over the network. The authors in [67] showed how spammers can guide spam towards specific regions, by leveraging only the public information available in OSNs.

A DOSN should include mechanisms able to limit the viral dissemination of advertisements or malicious software on peers of the system. In centralized OSNs, spam control is enforced by the centralized administration of the platform, but DOSNs require new distributed and decentralized methods for spam prevention. However, the characteristics of a DOSN, such as the limited network knowledge and the diffusion of the information on a unreliable network , make the implementation of the viral marketing more difficult.

7. DOSNs: current proposals

In this section we introduce a set of relevant DOSN proposals by classifying them according to the three classes introduced in Section 2: DHT-based, SO-based, or External resources-based.

7.1. DHT-based proposals

Table 2 summarizes the main proposals of DHT-based DOSNs and their characteristics. In the first four proposals reported in the table, the DHT is exploited to store social content, while in the last three proposals the DHT is exploited as an indexing service.

First of all, it is worth noticing that most proposals use the Pastry [68] DHT, and this may be due to the availability of FreePastry, an open-source stable implementation of Pastry. Only PeerSon [69,70] exploits OpenDHT, but the authors point out that it has been shown unreliable. Furthermore most proposals use the basic put/get DHT API for the management of social content, while only LifeSocial.KOM exploits a higher level service, i.e. PAST, built upon the DHT. PAST ia a distributed file system that guarantees both automatic data replication over the DHT, and retrieval of one of the stored copies. Finally, it is worth noticing that ABE is the encryption technique mostly used. As a matter of fact, ABE allows users to specify access control policies suitable for a social network. For instance, users can define policies based on the identities of the contacts, on their attributes and on groups of contacts.

We recall that the goal of all the proposals described in the following is to provide the basic functionalities of a OSN, but without the drawback of privacy loss. These functionalities should include, at least, the possibility to establish social links, to define a user's digital personal space, and communication channels supporting the sharing of social content among the users. In general, the DHT is used to support these functionalities.

DECENT [14] is an example of a DOSN fully based on a DHT which guarantees cryptographic protections for confidentiality and integrity. As shown in Section 3, DECENT defines container objects

Table 2
A comparison of DHT-based DOSNs

DOSN	year	DHT	Mobile devices	Storage	Diffusion	Privacy
PeerSon	2009	OpenDHT	yes	DHT	Request/Reply (DHT)	PKI
LifeSocial.KOM	2011	FreePastry	no	DHT - PAST	Request/Reply (DHT)	Broadcast Encryption
DECENT	2012	FreePastry/Kademlia	no	DHT	Request/Reply (DHT)	access control + ABE
Cachet	2012	FreePastry	no	DHT	Request/Reply (DHT + Gossip)	access control + ABE
My3	2011	not specified	no	Replication on Trusted Nodes (Trusted Proxy Set Nodes)	Request/Reply (through Trusted Proxy Set Nodes)	Encryption-based Access Control
GemStone	2011	Pastry	no	Data Holding Agents (may be untrusted)	Request/Reply	ABE encryption
SOUP	2014	FreePastry	yes	Mirror Nodes (may be untrusted)	Request/Reply	ABE encryption

that store both any type of social content, like profiles, posts, comments to posts, etc, and a list of comments/annotations. The container objects are mapped on the DHT using an object identifier as DHT key. In addition to the standard get and put requests, DECENT supports also an append operation, which is used to add a comment reference to an existing object. Data objects are encrypted by using an ABE encryption technique to provide confidentiality. Each object has three access policies associated with it: a read, a write and an append policy. Each policy may be attribute-based, identity based, or a combination of both. An example of policy is $((friend \wedge coworker) \vee family)$. The read policy is enforced through the use of ABE cryptographic protocols, while the write and append policies are enforced through a combination of cryptography and specialized DHT functionalities. To ensure a higher level of availability, several replicas are stored in the DHT, generally exploiting the neighbors of the node in the DHT. Each object is paired with a version number and a user can query all the replicas and use the freshest one.

PeerSon [69,70] has been one of the first proposal of DOSN, so it mainly deals with recreating the core features of OSNs in a decentralized way, in particular in the definition of a P2P storage system that takes into account the characteristics of OSN usage for storing social content. Advanced social networking features are not taken into account. PeerSon is a two-tier architecture in which one tier is implemented by a DHT and it serves as a look-up service. The second tier consists of peers and contains the user data, such as user profiles. The DHT stores only the meta-data required to search users and their data. Social data are stored on users devices that can directly exchange contents with other users. In PeerSon direct connection between devices is exploited. Indeed, communications between users are directly P2P when both are online, instead an asynchronous messaging support is available when users are not online at the same time.

LifeSocial.KOM [15] is a plugin-based architecture that provides the common functionalities of OSNs. An unique feature of LifeSocial.KOM is that its functionalities are implemented in individual software components, called plugins, which can be loaded and updated at run time. This plugin-based architecture enables an easy extension of the social network. LifeSocial.KOM uses a DHT, in detail FreePastry [71] and PAST [72] as data storage. Data are totally distributed and encrypted so that only authorized users can access data. Replication is used to provide data availability. The main characteristics of the last proposal where data are stored in the DHT, i.e. Cachet [58], has been presented in Section 5.1.3.

Let us now present some proposals that exploit the DHT as an indexing service. My3 [32] is a DHT-based DOSN which exploits the trust relationships among users to improve the availability of data in the network. Users' profiles are hosted only on a set of self-chosen trusted nodes, called Trusted Proxy Set (TPS). A user *u* and its set of trusted nodes (TPS) are stored in the DHT in the form of *(key,value)* pair, with *key* being the identifier of the user and *value* being references to the members of the *TPS*. This mapping is useful for contacting the nodes where the profile of a particular user is stored. The user trusts these nodes both for storing its profile content and for enforcing access control on the access requests. The selection of the TPS is guided by considering both the availability and the performance of the acquaintances as well as their geographical location.

Gemstone [21] acts as a middleware between social applications and the networking stack. It is a DHT-based DOSN in which the DHT stores routable identifiers of users, profile locations and information about the Data Holding Agents (DHAs), which represent a set of replica peers used to ensure data availability when the owners of data are offline. The choice of replicas nodes takes into account different selection strategies based on both temporal and social graph information.

SOUP [73] is based on the Pastry DHT and uses replication on mirrors nodes to increase data availability. The DHT stores the identifiers of mirror nodes. Note that, like most of the other approaches, SOUP use an ABE encryption to guarantee data confidentiality. The most interesting feature of SOUP is that it leverages social relationships in the mirror selection process. Each node *n* observes, for each friend *f* in the social graph, the behaviour of *n*'s mirrors, i.e. if the mirror has been able to provide *f*'s data. Each node periodically receives from its friend these observations about its mirrors and ranks them according to their reliability. The higher a mirror is ranked, the more likely it will make the node's data available.

Finally it is worth noticing that only a few DOSNs, like SOUP and Prometheus, which will be shown in the next section, take into account that most users exploit mobile devices to access the social network. Since these nodes may experience high churn and long response time, these DOSN consider these nodes as less stable when building the DHT. As a matter of fact, in SOUP, a mobile node is not directly joint to the DHT but it uses the DHT through a gateway node that is on the DHT.

7.2. SO-based proposals

Table 3 summarizes the most relevant characteristics of the SO-Based DOSNs. The most interesting feature of them is that all the proposals derive the social overlay from the knowledge of the social graph and of further specific information dynamically obtained by running the DOSN. For instance, in DiDuSoNet, the social overlay is defined by considering the Dunbar's circles of an ego and by defining direct connections only between users connected by a strong relationship, dynamically computed by considering their interactions. In Safebook, each node is the center of a matryoshka where the innermost shell is composed of nodes with direct trust relationships with the core node and each overlay hop connects a pair of nodes belonging to users linked by a trust relationship in real life.

Safebook [8] proposes a three tiers architecture for DOSN with the main focus on privacy, integrity and availability. Each user in SafeBook has a logical concentric structure called Matryoshka. Matryoshkas are concentric rings of nodes built around each user

Table 3
A comparison of SO-based DOSNs.

DOSN	Year	Social Overlay	Multiple devices	Storage	Diffusion	Privacy	DHT
SafeBook	2009	Matrioska	no	Replication on Trusted Nodes (in the innermost circle of the Matrioska)	Request/Reply (through trusted nodes)	PKI	-
Prometheus	2010	directed, labeled, and weighted multi-edged graph	yes	Replication on Trusted Nodes	Request/Reply (through Trusted Nodes)	Encryption-based Access Control	FreePastry
DiDuSoNet	2015	Dunbar-based	no	Replication on Trusted Nodes (2-replicas per node)	Request/Reply (through Trusted Nodes)	Trust Dunbar's nodes	Pastry

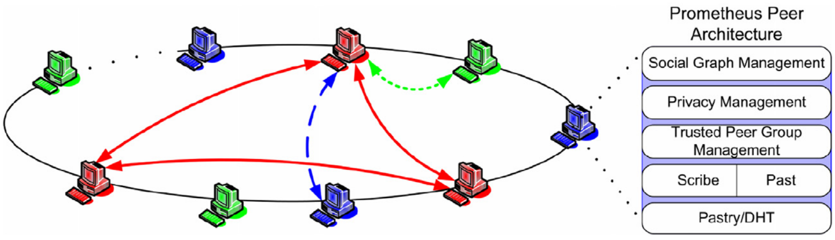


Fig. 6. Prometheus: system architecture.

that provide a trusted data storage and communication obfuscation through indirection. The ring structure is defined such as each user is at the center of a ring, and first ring contains the friend nodes that will host the encrypted user's data. Safebook takes also into consideration that, since the social network services pertaining to a user will be provided by this user's friend nodes, tracing of communications would easily disclose the friendship relationships in the social network. Safebook's solution is to protect the first ring with a second one consisting of nodes such that each one has a trusted contact with a node on the first ring. Further rings are built, recursively, through similar trust relationships. Each Matryoshka, defined by the concentric rings, protects the node, called *the core*, in its center, to preserve its privacy. The outermost circle of the Matryoshkas acts as a gateway for all the requests addressed to the core. Communication from a node outside the Matryoshka to the core passes through nodes in the Matryoshka which are connected through radial paths on which messages recursively can be relayed from the outermost shell to the core and vice versa. A complex update protocol is defined to automatically keep valid the Matryoshkas, even in case of dynamic nodes arrivals and departures.

DiDuSoNet [74] is a two-tier system, where the lower level is implemented by Pastry [68], and it is used for the bootstrapping phase, the look-up service, the search of other users, and to retrieve lists of replica nodes. The upper level is implemented by a Dunbar-based social overlay. DiDuSoNet exploits the Dunbar's approach [75] to define the social overlay. Dunbar explains that people have limited cognitive resources and, for this reason, each person can maintain a limited number of active social relationships. This limit, which is called the Dunbar's number, is 150. These 150 contacts can be arranged in concentric circles, where innermost circles includes contacts with whom a stronger social relationship is established. DiDuSoNet exploits the Dunbar's concept as a basis for the definition of the social overlay. The novelty of this system, with respect to the others described so far, is that it is completely based on trust between users, which is computed by the tie strength. In DiDuSonNet, the tie strength is calculated by taking into account different factors such as contact frequency between the two users, number of likes, posts, comments, private messages, tags. The social overlay is defined by connecting a user to all its Dunbar's contacts. Social data are stored only on trusted nodes, belonging to the Dunbur circles. Each node can choose two replicas, among its Dunbar friends, to have a high level of availability. The two replicas are chosen by considering different selection strategies that use a social score, as presented in [12], which takes into account the tie strength between nodes, the average session length, and the connection gain, as explained in Section 4.4.2. As soon as one replica becomes unavailable, the other is able to select another replica among the online nodes by exploiting the social score assigned to each node. The system manages both voluntarily and involuntarily disconnections of peers .

Prometheus [76] gathers information about users through *social sensors* that are applications running on behalf of the user. Social sensors are able to retrieve information from other sources, such as Facebook, and in particular they retrieve interactions with other users via email, phone, instant messaging, comments on blogs. These are used to create a weighted, directed, and labeled multi-edged graph, where vertices correspond to users and edges correspond to interactions between users. Fig. 6 shows the architecture of the system. Prometheus is built on top of Pastry [68] and it uses Past [72] as storage system. Social data are encrypted and can be stored at any peer. Each peer runs three Prometheus components for social graph management, privacy management, and trusted group management, as show in Fig. 6, where nodes with the same color are members of a users trusted peer group, instead red edges show group communication for social graph maintenance, while dashed blue and green edges show social inference queries submitted to other peers. Both the social information from sensors and the social subgraphs are stored and maintained in the P2P network. Information from sensors can be decrypted only by trusted peers, which are selected by users.

7.3. External Resource-based Proposals

Table 4 summarizes the characteristics of the main external-resource based DOSNs. Notice that most proposal exploit cloud storage as an external service. The basic idea is that cloud storage provides better availability than ordinary nodes storage. Furthermore, cloud utilities are generally trusted and do not claim any rights to the content placed on their infrastructure, like central servers of proprietary online social networks.

We will present each solution by exemplifying its key concepts before presenting the details of the solution.

Table 4
A comparison of External Resources-based DOSNs.

DOSN	Year	External Resource	Mobile devices	Storage	Diffusion	Privacy	P2P overlay
Vis-a-Vis	2011	Cloud	Yes	External Resources (Virtual Indivual Servers)	Request/Reply (Broadcast to Virtual Individual Servers)	Unencrypted data on trusted VISs	No
Vegas	2012	FTP, WebDAV, or a cloud storage	Yes	External Resources	Request/Reply (Broadcast to Datastores)	Link-specific key pair	Social Overlay based on relaxed Ego networks
SuperNova	2012	Cloud	No	Replication on Chosen Nodes + External Resources (Storekeepers)	Request/Reply (Broadcast to Storekeepers)	Access Control	Super-peers based network
Confidant	2011	Cloud services or existing OSNs	Yes	External Resources (Replication on servers controlled by trusted nodes)	Request/Reply (Broadcast)	Attribute-Based Access Control (ABAC)	
POSN	2015	Cloud	Yes	External Resources (Cloud Storage)	Request/Reply (Broadcast)	Public Key Encryption	Overlay Network between Mobile nodes
PrPl	2015	Cloud	Yes	External Resources (Replication on Personal Cloud Butlers)	Request/Reply (Broadcast)	Decentralized user centric identity management system	-
Diaspora	2010	Cloud	Yes	Servers (Pods)	Request/Reply (Broadcast)	SSL Encryption	-

An interesting proposal is that of Vis-a-Vis [77] whose main focus is the definition of a location-aware social network where users can join groups and share their location, at different granularity levels, with other members of the group.

Vis-a-Vis is a fully cloud-based proposal that exploits a federation of personal Virtual Individual Servers (VISs) which are virtual machines running in a paid cloud-computing environment. Users store their own data on their VIS and they could access the location-based groups through clients such as mobile applications and web browsers. The system supports group creation where the members of the group share a common interest in a particular topic. Location-based social networking is supported by defining, for each group, a *location tree* where higher nodes represent coarser geographic regions such as countries, while lower nodes represent finer regions such as cities. While the division of the map is predefined, users' geographic locations may be set statically (e.g., hometown) or updated dynamically (e.g., current GPS coordinates). Each leaf node corresponds to the VIS of a member of the group, regardless of the granularity at which members share their location, with the only constraint that the VIS must be a leaf node within the subtree covering its shared location. Each internal node of the tree is managed by its coordinator VIS, which is a VISs whose shared location is within the region corresponding to the internal node. Coordinators are elected through a distributed consensus protocol. The location tree is used to provide efficient, fault-tolerant, and scalable spatial range queries required for retrieving all users in a certain region.

Another interesting feature of Vis-a-Vis is that it is designed to inter-operate with existing OSNs such as Facebook. For instance, it is possible to integrate a Vis-a-Vis location-based group into a Facebook group by adding location information to the standard Facebook group.

Vegas [52] is a hybrid solution in which the DOSN takes advantage of external resources to increase the data availability. Vegas has a social overlay which is represented by ”a relaxed ego network” where each ego node knows about the direct connection with its friends, but it knows nothing about relationships between its friends or between its friends with unknown users. The visibility of the social graph is restricted to the relaxed ego network. A new friendship can be established only in two ways: by a face-to-face exchange of the public key or by exchanging it via encrypted NFC or Bluetooth connection, or by a simple protocol used only to create a friendship between two nodes with a common friend (a triangle). The system enables users to connect with mobile or stationary client devices and it introduces *datastores* for the management of data availability and for device synchronization. Datastores can be implemented as a web resource like FTP, WebDAV, or a cloud storage. In Vegas, each user holds a unique public key pair for each of its friends and messages are encrypted. When a user wants to send a message to a friend, it exchanges the specific public key with this friend and it chooses one or more exchangers. An exchanger represents the abstract concept of a message queue that is used to transmit messages. The system supports emailing, SMS, instant messaging and, micro-blogging (Twitter) as exchanger. This system suffers of several limitations: the creation of new friendships, the search operation to find other users that is not provided and also the communication channels used.

The main idea of SuperNova is to increase data availability by exploiting the heterogeneity of the nodes to define a set of super-peers and encourage them to offer their services to the community, by exploiting also a set of incentives.

In SuperNova [39], super-peers may be run either on end user computers, or on cloud services. They can help the bootstrap of new peers who do not yet have any friends, to maintain a directory of users to find friends by name or interests, and help peers to discover other peers. Furthermore super-peers store a user's content in case it does not have enough friends or if its friends are already overloaded. The incentives for a node to become a super-peer may be of various types. Super-peers get more insight in the network. This could be a self satisfaction for user, as a contributor in making the DOSN successful. Super-peer may spread advertisements, while serving other nodes content and be rewarded for this. Furthermore, peers contribute to the super-peers reputation, by rating the service they offer. Reputation is another incentive to improve their services.

When users have established enough relationships, they can select the storekeepers among their friends or also among further nodes of the network. Storekeepers are list of users who have agreed to keep a replication of another users data. When a node *n* is down, then the friends of *n* can contact storekeepers to access *n*s data. Data are stored in encrypted form at the super-peers and also at storekeepers.

Confidant's main ideas [16] are (1) to exploit the social graph as a guide to store users data on trusted nodes and (2) to define weakly consistent replication model to guarantee replica consistency. The idea of storing in plaintext on trusted nodes, is justified because the nodes of the users' friends already have read access to the user's social content, so they can be allowed to serve the user data on behalf of the user as well. Like in DiDuSoNet, the strength of social ties is inferred by the level of interaction between the users and is exploited to detect trusted friends. The replica set is synchronized through an epidemic propagation algorithm.

Users define groups of users, which are identified by names and are exploited to express access-control policies (ABAC) [78]. For example, a user may define groups with attributes such as âgcollege friends, or family.

Confidant relies on two low-cost infrastructures: storage servers and cloud-based name servers. A storage server hosts users' data and authenticates read and write requests. Users can select a small number of friends' servers to host replicas of their data and to authorize requests on behalf of the data owner, by executing their access-control policies. Name servers may be executed on free cloud services that are assumed to be highly available and are responsible for maintaining a list of available replicas (like a DNS) and for maintaining a logical clock used to enables replica synchronizion when replicas come back online.

POSN [79] is a proposal that totally relies on free of charge clouds for storing the encrypted data of the users. An interesting feature of this proposal is that it takes into consideration the security problems introduced by enabling users to directly modify the wall of their friends. In POSN, each user fetch friends' data from their personal cloud. Authors observe that if a friend is granted with write permission to write a comment on the wall, this might introduce security problems since he/she may intentionally or accidentally corrupt the file storing the wall's posts. POSN proposes to create a file for each specific friend that stores its comments. The wall may be built by considering the information of these files.

In POSN, a new relationship can be created only by using users' existing contacts, such as phone numbers or email addresses. Indeed, the platform can exchange emails or SMS messages between users to establish tokens that contain the cloud location and the public keys of friends. In order to provide security and privacy, data is encrypted before uploading into the cloud. Every user has a public key, that is exchanged during the friendship establishment, and shared through the cloud. When online, users can directly exchange information about common friends and common friends' latest posts to optimize the data distribution.

In PrPl [65] every user runs a person-centric service called Personal Cloud Butler, which is a service to organize user's data and share them with user's friends based on private preference, expressed by users. Larger content such as video, photo, music are stored in federated storages called Data Stewards. PrPl assumes that the butler service and users personal storage are deployed on devices with a high availability and for this reason, the system does not provide any replication or caching mechanism. PrPl relies on the OpenID [80] service for identity management. The PrPl system utilizes federated, decentralized identity management enabling secure logins, single sign-on, and communication among applications in an environment where Butlers belong to different administration domains.

The Diaspora network [81] is a network of independent, federated servers, that are administrated by users, which are used for storage, communication and access control. Users decide in which servers their information will be stored, and some users choose to maintain their own servers in order to keep complete control of their data. The private servers are called pods and they are dedicated resources to manage the reliable service provision [82]. For this reason, a pod can be hosted either on own hardware or by a cloud service. The main data in Diaspora are user profiles, posts comments, and messages, as Facebook. Users are represented by a profile that is stored using the hCard format[4]. Data are stored unencrypted on the servers, and the server administrator has access to the data. Communication between two users is managed by the pods they are registered with, and all messages are exchange by using the ssl encryption.

8. Conclusions and open problems

DOSNs represent a valid alternative to centralized OSNs. However, the decentralization requires specific approaches for the social data management. This paper presents an analysis of the existing DOSNs, focusing on the techniques to manage social contents in a distributed framework. In detail, the paper highlights three major problems for data management: data availability, information diffusion, and privacy. We describe in detail the problem of data availability by introducing a classification of the main techniques proposed to increase the availability of social contents. We introduce the problem of information diffusion and we classify current approaches into two major classes. We also describe the privacy and security problems arising in these systems. Moreover, we propose an overview of current DOSN proposals, first focusing on the main novel ideas they introduce, then giving a deeper presentation of their characteristics.

To conclude, we now discuss some limitations and open problems of current DOSNs that currently limit their widespread diffusion.

During the last years, DOSNs have been considered a new paradigm to implement social networks. However, despite the fact that Diaspora (one of the first and most popular examples of DOSN) has collected more than 400,000 users, other DOSNs have still limited impact. The main problem is that a few centralized social service providers like Facebook, managed by big companies, have dominated the OSNs market, attracting millions of users by offering advanced functionalities, and, consequently, people are reluctant to leave a network including so many contacts and advanced functionalities. Although this might change due to the recent emergence of huge data leaks, this problem does not seem so pressing for most users of Facebook, or other centralized OSNs.

In addition to acceptance issues, there are still technical issues that DOSN should solve to have the possibilty to make an impact. First of all DOSNs have still to solve several problems and offer the full range of functionalities of centralized OSNs. We describe the main problems that limit the diffusion of DOSN below:

Groups and Communities management. Groups and communities consist of an aggregation of users that share some common interests and their management is considered of great interest today. People join a community because they care about the common interests that glue the community members together. Some of them may join because they feel the urge to contribute to a cause sustained by the community; others join because they can benefit from being part of the community. Similar to a community, a group is a feature of many OSNs which allows users to share common interest. The management of groups/communities of users in DOSNs, still rises several challenges, in terms of information diffusion, and data management due to the dynamism of DOSNs. Indeed, there is the need of specific algorithms to manage communities or groups in dynamic environments, as described in [83]. A first contribution in this direction is proposed in [84], where a distributed protocol to detect and manage communities in a DOSN is proposed.

Privacy issues. Current DOSNs offer several mechanisms to address the privacy issues. However, there are still open problems. For example that of the impersonation attacks or the privacy

[4] http://microformats.org/wiki/hcard.

breaches coming from third-party applications accepted by users to get additional functionalities.

Monitoring the evolution of the social network. OSNs offer advanced functionalities to search and add new users to the friend list, like link prediction [85]. This is a useful tool both to understand the evolution of the network and to suggest relationships that may appear in a future network configuration. The prediction of a link (such as the link existence, type, its associated weight, etc.) is a classical problem in complex networks analysis [86]. However, DOSNs currently do not provide functionalities to suggest new friendship relations, because of the lack of distributed algorithms for the prediction of links.

Mobile communication. Nowadays, nearly 80% of all time spent on social media is from mobile nodes. At the same time, mobile phones are becoming more powerful, in particular in terms of battery life and connectivity. However, among existing DOSNs, only a few provide support for mobile access. PeerSoN enables direct exchange of data between devices to allow opportunistic and delay-tolerant networking. To support mobile users, SOUP minimizes data transfer and resource consumption on mobile nodes. To summarize, even if a support for mobile communication is really important, it is still an open problem for most DOSNs.

Trust vs. Encryption Most existing DOSNs use classic encryption techniques (symmetric or asymmetric encryption), to guarantee privacy and anonymity. Encryption adds some overhead in both the time and space domains. In [87], authors propose an analysis of the encryption cost during both join and leave operations in DHT-based DOSNs. However, one of the major cost concerns the decryption of content during users' requests. Contents are often requested and modified by users, for example when a post is published or a comment to a post is inserted. In [60], authors present a detailed analysis of the cost of encryption in terms of storage cost, encryption/decryption time, transmission cost and, the management of groups. An alternative solution is to avoid encryption by storing content, in clear, on trusted nodes. Trust is widely accepted as a major component of human social relationships and is a characteristic of real life where people rely on trusted persons. In general, trust is a measure of confidence that can be evaluated by analysing the interactions between users. DiDuSoNet [74] and Confidant [16] are the only DOSNs in which encryption is not used and data are stored, in clear, on trusted node.

The automatic evaluation of trustworthiness can be considered still an open problem strictly tied to the evaluation the nature of an interaction of two users. We think that an approach combining encryption and trust could be a good solution that overcome the main limitations of both of them when they are used individually.

References

[1] D. Boyd, N.B. Ellison, Social network sites: definition, history, and scholarship, J. Comput.-Med. Commun. 13 (1–2) (2007).

[2] M. Kryczka, R. Cuevas, C. Guerrero, E. Yoneki, A. Azcorra, A first step towards user assisted online social networks, in: Proceedings of the 3rd Workshop on Social Network Systems, SNS '10, 2010, pp. 6:1–6:6.

[3] A. Datta, S. Buchegger, L.-H. Vu, T. Strufe, K. Rzadca, Decentralized online social networks, in: Handbook of Social Network Technologies and Applications, 2010, pp. 349–378.

[4] J. Buford, H. Yu, E.K. Lua, P2P Networking and Applications, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

[5] Diaspora. https://joindiaspora.com,

[6] T. Paul, A. Famulari, T. Strufe, A survey on decentralized online social networks, Comput. Netw. 75, Part A (0) (2014) 437–452.

[7] S.R. Chowdhury, A.R. Roy, M. Shaikh, K. Daudjee, A taxonomy of decentralized online social networks, Peer-to-Peer Netw. Appl. 8 (3) (2015) 367–383.

[8] L.A. Cutillo, R. Molva, T. Strufe, Safebook: a privacy-preserving online social network leveraging on real-life trust, Comm. Mag. 47 (12) (2009) 94–101.

[9] A.S. Tanenbaum, M. Van Steen, Distributed Systems: Principles and Paradigms, Prentice-Hall, 2007.

[10] G. Mega, A. Montresor, G.P. Picco, Efficient dissemination in decentralized social networks., in: Peer-to-Peer Computing, IEEE, 2011, pp. 338–347.

[11] P. Marsden, Egocentric and sociocentric measures of network centrality, Soc. Netw. 24 (4) (2002) 407–422.

[12] M. Conti, A. De Salve, B. Guidi, F. Pitto, L. Ricci, Trusted dynamic storage for dunbar-based p2p online social networks, in: Proceedings of the OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Springer, 2014a, pp. 400–417.

[13] A. De Salve, P. Mori, L. Ricci, R. Al-Aaridhi, K. Graffi, Privacy-preserving data allocation in decentralized online social networks, in: Distributed Applications and Interoperable Systems, 2016d, pp. 47–60.

[14] S. Jahid, S. Nilizadeh, P. Mittal, N. Borisov, A. Kapadia, DECENT: A decentralized architecture for enforcing privacy in online social networks, in: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on, IEEE, 2012, pp. 326–332.

[15] K. Graffi, C. Gross, D. Stingl, D. Hartung, A. Kovacevic, R. Steinmetz, Lifesocial. kom: a secure and p2p-based solution for online social networks, in: Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC), IEEE, 2011, pp. 554–558.

[16] D. Liu, A. Shakimov, R. Cáceres, A. Varshavsky, L.P. Cox, Confidant: protecting OSN data without locking it up, in: Proceedings of the 12th International Middleware Conference, International Federation for Information Processing, 2011, pp. 60–79.

[17] L. Yin, G. Cao, Supporting cooperative caching in ad hoc networks, IEEE Trans. Mobile Comput. 5 (2006) 77–89. http://doi.ieeecomputersociety.org/10.1109/TMC.2006.15.

[18] D.P. John, J. Harrison, A distributed internet cache, in: Proceedings of the 20th Australian Computer Science Conference, 1997, pp. 5–7.

[19] S. Acharya, S.B. Zdonik, An efficient scheme for dynamic data replication, Technical Report, 1993. Providence, RI, USA

[20] R. Rodrigues, B. Liskov, High Availability in DHTs: Erasure Coding vs. Replication, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 226–239.

[21] F. Tegeler, D. Koll, X. Fu, Gemstone: empowering decentralized social networking with high data availability, in: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2011), IEEE, 2011, pp. 1–6.

[22] L. Gyarmati, T.A. Trinh, Measuring user behavior in online social networks, Netw. IEEE 24 (5) (2010) 26–31.

[23] J. Jiang, C. Wilson, X. Wang, P. Huang, W. Sha, Y. Dai, B.Y. Zhao, Understanding latent interactions in online social networks, in: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, in: IMC '10, 2010, pp. 369–382.

[24] Z. Xu, R. Min, Y. Hu, Reducing maintenance overhead in DHT based peer-to-peer algorithms, in: Proceedings of the Third International Conference on Peer-to-Peer Computing, 2003.(P2P 2003)., IEEE, 2003, pp. 218–219.

[25] A. De Salve, M. Dondio, B. Guidi, L. Ricci, The impact of user's availability on on-line ego networks, Comput. Commun. 73 (PB) (2016b) 211–218.

[26] A. De Salve, B. Guidi, P. Mori, L. Ricci, V. Ambriola, Privacy and temporal aware allocation of data in decentralized online social networks, in: Proceedings of the International Conference on Green, Pervasive, and Cloud Computing, 2017, pp. 237–251.

[27] R. Sharma, A. Datta, M. Dell Amico, P. Michiardi, An empirical study of availability in friend-to-friend storage systems, in: Proceedings of the P2P 2011, IEEE International Conference on Peer-to-Peer Computing, 2nd, 2011, Kyoto, Japan, 2011, pp. 348–351.

[28] R. Bhagwan, S. Savage, G.M. Voelker, Understanding availability, in: Peer-to-Peer Systems II, Springer, 2003, pp. 256–267.

[29] R. Gracia-Tinedo, M. S'nchez-Artigas, P. Garcia-Lopez, F2box: cloudifying f2f storage systems with high availability correlation, in: Proceedings of the IEEE 5th International Conference on Cloud Computing (CLOUD), IEEE, 2012, pp. 123–130.

[30] S. Fu, L. He, X. Liao, C. Huang, Developing the cloud-integrated data replication framework in decentralized online social networks, J. Comput. Syst. Sci. 82 (1) (2016) 113–129.

[31] A. Shakimov, A. Varshavsky, L.P. Cox, R. Cáceres, Privacy, cost, and availability tradeoffs in decentralized OSNS, in: Proceedings of the 2nd ACM Workshop on Online Social Networks, ACM, 2009, pp. 13–18.

[32] R. Narendula, A. Papaioannou, K. Aberer, A decentralized online social network with efficient user-driven replication, in: Proceedings of the IEEE International conference on Social Computing (SocialCom 2012), 2012a.

[33] D. Schiöberg, F. Schneider, G. Tredan, S. Uhlig, A. Feldmann, Revisiting content availability in distributed online social networks, CoRR (2012) arXiv:1210.1394.

[34] A. Olteanu, G. Pierre, Towards robust and scalable peer-to-peer social networks, in: Proceedings of the Fifth Workshop on Social Network Systems, ACM, 2012, p. 10.

[35] J. Li, F. Dabek, F2F: reliable storage in open networks, in: Proceedings of the 5th International workshop on Peer-To-Peer Systems, IPTPS 2006, Santa Barbara, CA, USA, 2006.

[36] A. De Salve, B. Guidi, P. Mori, L. Ricci, Distributed coverage of ego networks in f2f online social networks, in: Proceedings of the International IEEE Conferences Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), IEEE, 2016c, pp. 423–431.

[37] R. Narendula, T.G. Papaioannou, K. Aberer, Towards the realization of decentralized online social networks: an empirical study., in: Proceedings of the ICDCS Workshops, IEEE Computer Society, 2012b, pp. 155–162.

[38] M. Dell'Amico, M. Filippone, P. Michiardi, Y. Roudier, On user availability prediction and network applications, IEEE/ACM Trans. Netw. 23 (4) (2015) 1300–1313.

[39] R. Sharma, A. Datta, Supernova: Super-peers based architecture for decentralized online social networks, in: Proceedings of the COMSNETS, 2012, pp. 1–10.
[40] P.T. Eugster, P.A. Felber, R. Guerraoui, A.-M. Kermarrec, The many faces of publish/subscribe, ACM Comput. Surv. 35 (2) (2003) 114–131.
[41] P. Triantafillou, I. Aekaterinidis, Content-based publish-subscribe over structured p2p networks, in: Proceedings of the 4th International Workshop on Distributed Event-Based Systems, 30, 2004.
[42] A. Datta, S. Quarteroni, K. Aberer, Autonomous gossiping: a self-organizing epidemic algorithm for selective information dissemination in wireless mobile adhoc networks, pp. 126–143.
[43] P.T. Eugster, R. Guerraoui, A.-M. Kermarrec, L. Massoulié, Epidemic information dissemination in distributed systems, Computer 37 (5) (2004) 60–67.
[44] A. Lindgren, A. Doria, O. Schelen, Probabilistic routing in intermittently connected networks, Service Assurance with Partial and Intermittent Resources, 2004, pp. 239–254.
[45] J. Li, Y. Chen, Z. Lin, W. Chen, B. Vucetic, L. Hanzo, Distributed caching for data dissemination in the downlink of heterogeneous networks, IEEE Trans. Commun. 63 (10) (2015) 3553–3568.
[46] P. Ruiz, P. Bouvry, Survey on broadcast algorithms for mobile ad hoc networks, ACM Comput. Surv. 48 (1) (2015) 8:1–8:35.
[47] J. Kangasharju, J. Roberts, K.W. Ross, Object replication strategies in content distribution networks, Comput. Commun. 25 (4) (2002) 376–383.
[48] M. Bender, S. Michel, P. Parkitny, G. Weikum, A comparative study of pub/sub methods in structured p2p networks, in: Databases, Information Systems, and Peer-to-Peer Computing, Springer, 2007, pp. 385–396.
[49] G. Mega, A. Montresor, G.P. Picco, Cloud-assisted dissemination in social overlays, in: Proceedings of the IEEE Thirteenth International Conference on Peer–to-Peer Computing (P2P), IEEE, 2013, pp. 1–5.
[50] L. Han, B. Nath, L. Iftode, S. Muthukrishnan, Social butterfly: social caches for distributed social networks, in: Proceedings of the SocialCom/PASSAT, 2011, pp. 81–86.
[51] L. Han, M. Punceva, B. Nath, S. Muthukrishnan, L. Iftode, Socialcdn: caching techniques for distributed social networks, in: Proceedings of the P2P, 2012, pp. 191–202.
[52] M. Durr, M. Maier, F. Dorfmeister, Vegas–a secure and privacy-preserving peer–to-peer online social network, in: Proceedings of the International Conference on and 2012 International Confernece on Social Computing (SocialCom) Privacy, Security, Risk and Trust (PASSAT), 2012, pp. 868–874.
[53] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, D. Terry, Epidemic algorithms for replicated database maintenance, in: Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing, in: PODC '87, 1987, pp. 1–12.
[54] M. Conti, A. De Salve, B. Guidi, L. Ricci, Epidemic diffusion of social updates in dunbar-based dosn., in: Proceedings of the Euro-Par Workshops (1), 2014b, pp. 311–322.
[55] A. Datta, R. Sharma, GoDisco: selective gossip based dissemination of information in social community based overlays, in: Proceedings of the 12th international conference on Distributed computing and networking, in: ICDCN'11, 2011, pp. 227–238.
[56] R. Sharma, A. Datta, GoDisco++: A gossip algorithm for information dissemination in multi-dimensional community networks, Pervasive Mob. Comput. 9 (2) (2013) 324–335.
[57] U. Tandukar, J. Vassileva, Selective propagation of social data in decentralized online social network, in: Advances in User Modeling, Springer, 2012, pp. 213–224.
[58] S. Nilizadeh, S. Jahid, P. Mittal, N. Borisov, A. Kapadia, Cachet: a decentralized architecture for privacy preserving social networking with caching, in: Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies, in: CoNEXT '12, ACM, 2012, pp. 337–348.
[59] H. Gao, J. Hu, T. Huang, J. Wang, Y. Chen, Security issues in online social networks, Internet Comput. IEEE 15 (4) (2011) 56–63.
[60] O. Bodriagov, S. Buchegger, Encryption for peer-to-peer social networks, in: Security and Privacy in Social Networks, 2013, pp. 47–65.
[61] O. Bodriagov, G. Kreitz, S. Buchegger, Access control in decentralized online social networks: applying a policy-hiding cryptographic scheme and evaluating its performance, in: Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), IEEE, 2014, pp. 622–628.
[62] J. Katz, A. Sahai, B. Waters, Predicate encryption supporting disjunctions, polynomial equations, and inner products, in: Advances in Cryptology–EUROCRYPT 2008, Springer, 2008, pp. 146–162.
[63] L.-H. Vu, K. Aberer, S. Buchegger, A. Datta, Enabling secure secret sharing in distributed online social networks, in: Proceedings of the Computer Security Applications Conference, 2009. ACSAC'09. Annual, IEEE, 2009, pp. 419–428.
[64] B. Greschbach, G. Kreitz, S. Buchegger, The devil is in the metadata-new privacy challenges in decentralised online social networks, in: Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), IEEE, 2012, pp. 333–339.
[65] S.-W. Seong, J. Seo, M. Nasielski, D. Sengupta, S. Hangal, S.K. Teh, R. Chu, B. Dodson, M.S. Lam, Prpl: a decentralized social networking infrastructure, in: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing and Services: Social Networks and Beyond, in: MCS '10, 2010, pp. 8:1–8:8.
[66] J. Leskovec, L.A. Adamic, B.A. Huberman, The dynamics of viral marketing, in: EC '06: Proceedings of the 7th ACM Conference on Electronic Commerce, ACM Press, 2006, pp. 228–237.
[67] G. Brown, T. Howe, M. Ihbe, A. Prakash, K. Borders, Social networks and context-aware spam, in: Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work, ACM, 2008, pp. 403–412.
[68] A. Rowstron, P. Druschel, Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems 2218 (2001) 329–350.
[69] S. Buchegger, D. Schioberg, L. Vu, A. Datta, Implementing a P2P social network - early experiences and insights from peerSoN, in: Proceedings of the Second ACM Workshop on Social Network Systems (Co-located with EuroSys 2009),
[70] S. Buchegger, D. Schiöberg, L.-H. Vu, A. Datta, Peerson: P2p social networking: early experiences and insights, in: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, ACM, 2009, pp. 46–52.
[71] Freepastry. http://www.freepastry.org/freepastry,
[72] P. Druschel, Past: A large-scale, persistent peer-to-peer storage utility, in: Proceedings of the HotOS VIII, 2001, pp. 75–80.
[73] D. Koll, J. Li, X. Fu, Soup: an online social network by the people, for the people, in: Proceedings of the 15th International Middleware Conference, ACM, 2014, pp. 193–204.
[74] B. Guidi, T. Amft, A. De Salve, K. Graffi, L. Ricci, Didusonet: a p2p architecture for distributed dunbar-based social networks, Peer-to-Peer Netw. Appl. 9 (6) (2016) 1177–1194.
[75] R.I.M. Dunbar, The social brain hypothesis, Evolut. Anthropol.: Issues, News Rev. 6 (5) (1998) 178–190.
[76] N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, A. Iamnitchi, Prometheus: user-controlled p2p social data management for socially-aware applications, in: Proceedings of the ACM/IFIP/USENIX 11th International Conference on Middleware, in: Middleware '10, Springer-Verlag, 2010, pp. 212–231.
[77] A. Shakimov, H. Lim, R. Cáceres, L.P. Cox, K. Li, D. Liu, A. Varshavsky, Vis-a-vis: privacy-preserving online social networking via virtual individual servers, in: Proceedings of the Third International Conference on Communication Systems and Networks (COMSNETS), 2011, pp. 1–10.
[78] V.C. Hu, D.R. Kuhn, D.F. Ferraiolo, Attribute-based access control, Computer 48 (2) (2015) 85–88.
[79] E. Erdin, E. Klukovich, G. Gunduz, M.H. Gunes, Posn: a personal online social network, in: Proceedings of the IFIP International Information Security Conference, Springer, 2015, pp. 51–66.
[80] D. Recordon, D. Reed, Openid 2.0: A platform for user-centric identity management, in: Proceedings of the Second ACM Workshop on Digital Identity Management, in: DIM '06, 2006, pp. 11–16.
[81] A. Bielenberg, L. Helm, A. Gentilucci, D. Stefanescu, H. Zhang, The growth of diaspora-a decentralized online social network in the wild, in: Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2012, pp. 13–18.
[82] S. Schulz, T. Strufe, d2 deleting diaspora: practical attacks for profile discovery and deletion, in: Proceedings of the 2013 IEEE International Conference on Communications (ICC), 2013, pp. 2042–2046.
[83] B. Guidi, A. Michienzi, G. Rossetti, Dynamic community analysis in decentralized online social networks, in: Proceedings of the Euro-Par 2017: Parallel Processing Workshops, Springer International Publishing, 2018b, pp. 517–528.
[84] B. Guidi, A. Michienzi, L. Ricci, Sonic-man: a distributed protocol for dynamic community detection and management, in: Distributed Applications and Interoperable Systems, 2018a, pp. 93–109.
[85] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, J. Assoc. Inf. Sci. Technol. 58 (7) (2007) 1019–1031.
[86] L. Lü, T. Zhou, Physica A: Statistical Mechanics and its Applications.
[87] A. De Salve, R. Di Pietro, P. Mori, L. Ricci, Logical key hierarchy for groups management in distributed online social network, in: Proceedings of the IEEE Symposium on Computers and Communication (ISCC), IEEE, 2016a, pp. 710–717.

Barbara Guidi is currently a postdoctoral researcher at the Department of Computer Science of the University of Pisa. She received her B.Sc. and M.Sc. in Computer Science from the University of Pisa, Italy, in 2007 and 2011, respectively. She received her Ph.D. degree in Computer Science from the University of Pisa, in 2015. In 2014, during her Ph.D., she was a visitor at the Heinrich Heine University of Dusseldorf. She was a Co-Chair for the conference EAI GoodTechs 2017, and she is a workshop Co-Chair for LSDVE 2018, in conjunction with EUROPAR 2018.

She received two Best Paper Awards: at the International Conference DCNET 2013 and at the workshop LSDVE 2017.

She has been involved in the TPC of conferences and workshops, such as IEEE CloudCom, and has been a reviewer for journals, such as Mobile Networks and Applications (MONET), Multimedia Tools and Applications (MTAP), and Concurrency and Computation: Practice and Experience (CCPE). Her current research interests include distributed systems, P2P networks, complex networks, Decentralized Online Social Networks, dynamic community detection, and the blockchain technology.

Marco Conti is a research director and scientific counselor, for information and communication technologies, of the Italian National Research Council. He has published in journals and conference proceedings more than 400 scientific papers related to design, modelling, and experimentation of Internet architecture and protocols, pervasive systems and social networks. He has published the books “Metropolitan Area Networks (MANs)” (1997), “Mobile Ad Hoc Networking” (2004), “Mobile Ad hoc networking: the cutting edge technologies” (2013) and “Online Social Networks: Human Cognitive Constraints in Facebook and Twitter Personal Graphs” (2015). He has received several awards, including the Best Paper Award at IFIP TC6 Networking 2011, IEEE ISCC 2012 and IEEE WoWMoM 2013. He is Editor-in-Chief of Computer Communications and Online Social Networks and Media, and Editor-in-Chief for special issues of Pervasive and Mobile Computing. He served as TPC chair for several major conferences, such as IFIP Networking 2002, IEEE WoWMoM 2005, IEEE PerCom 2006, and ACM MobiHoc 2006, and he was general chair (among many others) for IEEE WoWMoM 2006, IEEE MASS 2007 and IEEE PerCom 2010.

He is the founder of successful conference and workshop series, such as IEEE AOC, ACM MobiOpp, and IFIP SustainIT.

Andrea Passarella (PhD 2005) is currently a Researcher at the Institute for Informatics and Telematics (IIT) of the National Research Council of Italy (CNR). Prior to join IIT he was with the Computer Laboratory of the University of Cambridge, UK. He has published 130+ papers on various topics, including human-centric data management for self-organising networks, Online and Mobile social networks, opportunistic, ad hoc and sensor networks. He received four best paper awards, including at IFIP Networking 2011 and IEEE WoWMoM 2013. He is General Co-Chair for IEEE WoWMoM 2019 and workshops co-chair for IEEE INFOCOM 2019. He was the PC co-chair of IEEE WoWMoM 2011, Workshops co-chair of ACM MobiHoc 2015, IEEE PerCom and WoWMoM 2010, and the co-chair of several IEEE and ACM workshops.

He is co-author of the book “Online Social Networks: Human Cognitive Constraints in Facebook and Twitter Personal Graphs” (Elsevier, 2015), and was Guest Co-Editor of the special issue Online Social Networks in Elsevier Computer Communications, and several special sections in ACM and Elsevier Journals and of the book “Multi-hop Ad hoc Networks: From Theory to Reality” (2007). He is the chair of the IFIP WG 6.3 “Performance of Communication Systems”.

Laura Ricci received the M. Computer Science from the University of Pisa in 1983 and the Ph.D. from the University of Pisa in 1990. Currently, she is an Assistant Professor at the Department of Computer Science, University of Pisa, Italy. Her research interests include parallel and distributed systems, peer-to-peer networks, cryptocurrencies and blockchains.

In this field, she has co-authored over 100 papers in refereed scientific journals and conference proceedings. She has served as a program committee member of several conferences and has been a reviewer for several journals.