Privacy Analysis on Microblogging Online Social Networks: a Survey

Samia Oukemeni[1], Helena Rifà-Pous[2], Joan Manuel Marquès Puig[3]

Abstract
In the last ten years, Online Social Networks (OSNs) embrace many different forms of interactive communication, including multimedia sharing, microblogging services, etc. They allow users to create profiles, connect with friends and share their daily activities and thoughts. However, this ease of use of OSNs come with a cost in terms of users' privacy and security. The big amount of personal data shared in the users' profiles or correlated from their activities can be stored, processed and sold for advertisement or statistical purposes. It attracts also malicious users who can collect and exploit the data and target different types of attacks. In this paper, we review the state of the art of OSNs existing either in the literature or deployed for use. We focus on the OSN systems that offer, but not exclusively, microblogging services. We analyze and evaluate each system based on a set of characteristics, and we compare them based on their usability and the level of protection of privacy and security they provide. This study is a first step towards understanding the security and privacy controls and measuring their level in an OSN.

Keywords
Online social networks (OSNs) – Microblogging systems — Social network security and privacy – Privacy protections – Usability in security and privacy

[1] *soukemeni@uoc.edu, Universitat Oberta de Catalunya, Barcelona, Spain*
[2] *hrifa@uoc.edu, Universitat Oberta de Catalunya, Barcelona, Spain*
[3] *jmarquesp@uoc.edu, Universitat Oberta de Catalunya, Barcelona, Spain*

Contents

1. Introduction

Since 2004, OSNs, especially microblogging services such as Twitter, have grown and gained a notorious popularity among users. They have penetrated daily life, connecting people all over the world. In fact, the number of users in online social services exceeded 2 billion monthly active users between Facebook, Twitter and Tumblr, in the 1st quarter of 2017 [1, 2, 3]. Initially, people used the services of social networks to connect with friends and share interests via short messages. However, the social changes and political movements have added a new role to OSNs. They have become a source of news coverage and means of propagating all sorts of information. Some examples are the "Arab Spring" movement in the Middle East and North Africa (MENA) in 2011 [4] or

the coverage of the 2016 US presidential election [5].

OSNs offer various functionalities and services that attract a great number of users to online social services. The users are instantly informed of news of their interests and their entourage. In addition, OSNs can analyze data and correlate users' interests to give advanced and personalized services. They can recommend potential friends or interests based on the information extracted from the users' profiles and activities (preferences, daily browsing, etc.) as well as from the their followers' activities. However, having the OSN services managed by a single provider entitles them to some risks of availability and privacy. The Internet shutdowns and servers' failures can be a bottleneck to the traffic and make the services completely unavailable to the users like what happened recently in May 2017, when WhatsApp services shut down for several hours and blocked all means of communication [6]. Besides, having a system provider that collects, stores, correlates and sells users' information and interests can threaten the user's privacy and lead to cyberattack.

Current studies [7, 8] have shown that many users jeopardize their private life by posting sensitive information, such as photos, phone numbers or locations. For example, the website Please Rob Me [9] scans Twitter feeds and shows when the users tweet out locations other than their home (the purpose of this site is to raise the awareness of the danger of location-based services). According to Hallinan et al. [10], a great number of users of online systems gives a high value to the protection of privacy and they understand the risk of releasing private data. They have the constant fear and the uncertainty of what happens to their private data once it is released. But at the same time, they accept to release data as a price to live in a modern world. This contradiction between giving the consent to process and sell personal data and the concern about the protection of privacy is called "the privacy paradox" [11].

While it is true that users can adjust the privacy settings provided by the services and limit the access to their profiles and contents, these controls remain insufficient to fully protect the users. During the entire registration phase, privacy policies are hidden in "Terms of Services" which are provided as an external link and ignored most of the time [12]. Furthermore, service providers retain the right to change the clauses of the policies at any time. Additionally, users give implicitly their full consent to service providers to store, process and analyze their data and sometimes sell it to third parties for advertising and marketing purposes. In addition, the service providers control the databases where the users' data are stored. This power of control makes the act of censorship easily performed.

The cited issues have motivated researchers to propose and build different privacy-preserving online social systems that protect the users' privacy, anonymity and confidentiality and, at the same, they are secure and censorship resistant. These systems differ in the features they provide, the architecture and the degree of privacy and security protection offered to users. Some researchers have maintained the central architecture of the systems, but they boost their solutions with techniques to protect users' privacy, like Hummingbird [13]. Others have proposed wrappers around some known existing platforms, like Twitter, in order to preserve the commercial interest of such systems and, at the same time, provide privacy for users, for example, Twitsper [14]. Other projects have proposed decentralized OSNs like Diaspora [15].

This variety of solutions of OSNs was a motivation to study and compare the different previous works and projects. Some OSNs are deployed for use, some others are only proposed in the literature as proof of concepts and prototypes. The primary scope of the present survey is online social systems that provide, but not exclusively, microblogging services with a focus on the techniques used to preserve the privacy and security of the users.

According to the literature, some papers have already proposed comparative surveys based on privacy-preserving features in OSNs. These surveys generally focus their discussion on decentralized OSNs or on just one aspect of social networks, either based on the features the system provides or based on the security and privacy issues in OSNs.

In [16], Paul et al. focus on comparing 16 decentralized OSNs. They divide the systems into three categories based on the types of decentralized storage of content: storing data (1) on peer nodes, (2) on external servers (federated-OSNs) or (3) on a hybrid of both types. Then, the authors compare the systems based on the access control approaches used and based on the way how interaction and signaling mechanisms are implemented. In [17], Chowdhury et al. create a taxonomy of decentralized OSNs and compare eight different decentralized microblogging systems based on their architecture (structured or unstructured), the types of service they provide (read only or read-write services), social application development API, the availability architecture, the scalability, the privacy control, security model, and the business model.

Some other articles in the literature focus on security and privacy issues of OSNs. NaliniPriya et al. [18] highlight security issues and possible attacks on OSNs. The authors classify the attacks under several categories: (1) classic threats like malware, phishing attacks and spammers, (2) modern threats such as fake profiles, location, and information leakage or clickjacking, and (3) adolescent attacks like cyber-bullying and stalking. In summary, although there is a rich literature in the area, current studies still lack analyzing more criteria (such as censorship resistance or anonymity and unlinkability,...) to evaluate the OSNs and to address all privacy preserving techniques used in online social systems. The present survey addresses these issues providing a comprehensive comparison of OSNs and giving the scientific community a knowledge base to understand OSNs based on different aspects. This survey considers the relationship between privacy, security, and usability, in particular, the "trade-off" paradigm where an increase of privacy and security protections would inevitably influence the friendliness and the usability of the system.

The rest of this survey is organized as follows. In section 2, we present the set of features and characteristics we have selected

to compare the OSNs. We introduce and briefly describe a selection of different OSNs in section 3. Section 4 is dedicated to comparing the different projects and systems based on the set of characteristics described in section 2 and to elaborating the impact of security and privacy preserving techniques on the usability and user friendliness of OSNs. Section 5 presents a conclusion of the present paper with some deductions and lines for future works.

2. Features and Characteristics

To understand how different OSNs operate and address the issues of the privacy and security, we identified seven main criteria for classification and comparison as shown in figure 1: (1) the type of service provided, (2) the architecture, (3) the storage and replication techniques, (4) the encryption mechanisms and key management, (5) the functionalities, (6) the security goals, (7) and the privacy goals. In this section, we discuss the set of characteristics we have identified to evaluate and compare different OSNs.

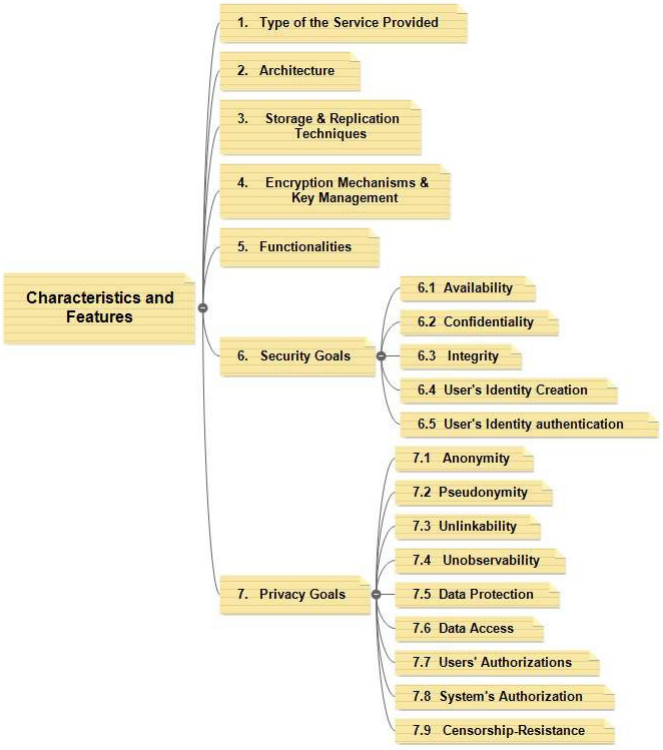

Figure 1. List of Features and Characteristics.

2.1 Type of the Service Provided

The first criterion to base on the comparison of different OSNs is the type of the service provided by the system. In addition to microblogging services, OSNs offer different types of services like multimedia sharing, social review, online chatting, etc.

2.2 Architecture

Online social systems adopt 3 different types of architecture: (1) centralized architecture, (2) decentralized architecture (federated or totally decentralized) , or (3) hybrid architecture that combines elements from both of the previous architectures.

2.3 Storage and Replication Techniques

The storage of data in OSNs differs from a system to another. In general, there are 4 methods of storing data: (1) on centralized services maintained by a single authority, (2) on federated servers with multiple authorities, (3) on decentralized services, and (4) on a hybrid of centralized and decentralized services where some forms of data are stored on the nodes and other forms are stored on centralized services. Replication mechanisms are used as a guarantee of availability and accessibility of the service and the data when needed.

2.4 Encryption Mechanisms and Key Management

Another criterion to compare OSNs is the encryption mechanisms and the cryptographic key exchange management used in the system. They are used to provide confidentiality. OSNs rely on different types of cryptographic algorithms: (1) symmetric algorithms with shared-keys known to all stakeholders, and (2) asymmetric or public-key algorithms with a pair of keys (public and private keys).

2.5 Functionalities

Based on the type of the service provided, OSNs offer different sets of functionalities to their clients. We have identified 13 functionalities that a user can perform on an OSN. Each functionality presents a risk of some degree on the privacy of users.

- Profile management (create, edit and delete profiles): the users can choose what information to publish on the profile.
- Visibility of the profile to others (followers, users, service provider...): personal information about the users can be extracted from the profile.
- Relationship handling: add, accept and remove followers and the visibility of the relationship to others (followers, users, service provider...).
- Follow interests (#Hashtags): this can reveal the interest of the users.
- Mentioning (User): this can reveal the social graph of the users.
- Reshare of a post: this can reveal the identity of the first publisher which makes the private messages public.
- Reply and comment others' posts: this can reveal the social graph of the users.
- Search function: find other users, word search, search for comments, etc...
- Recommendation of content or users: these recommendations are usually generated from analyzing the activities and interests of users.

- Content sharing: message shared with the public, a group or just one person.
- Content visibility: the visibility of message to the public, a group or just one person and the visibility of the message content to the service provider.
- Instant messaging: one-to-one or group online chatting.
- Media sharing (the ability to share files, videos, photos, links, etc...): reveals the interest of the users.

2.6 Security Goals

The protection of the security of OSNs relies mainly on the AIC triad (availability, integrity and confidentiality) [19, 20]. We added two more criteria to understand how the surveyed system handles the users' identity creation and authentication.

1. **Availability** ensures the access to authorized data and resources at any time and from everywhere.
2. **Confidentiality** protects the data content and prevents any unauthorized disclosure.
3. **Integrity** ensures the reliability of the data, stored or in transit, and guarantees that any unauthorized modification is blocked.
4. **User's identity creation** and registration in the system.
5. **User's identity authorization and authentication** in the system.

2.7 Privacy Goals

To evaluate the privacy of an OSN, we combined two models that address privacy-specific goals in systems: LINDDUN [21] and PriS Method [22]. We identified 9 goals to protect data privacy in an OSN. Some privacy goals in the identified set are overlapping with security goals; however, they are included in the set considering their importance in the protection of privacy. The following defines briefly each property:

1. **Anonymity** can be defined as the impossibility of identifying a subject within a set of subjects [23]. For example, when an anonymous message is received, it is difficult to link it to its sender.
2. **Pseudonymity** refers to using a pseudonym as an identifier of a subject other than the real name [23].
3. **Unlinkability** is defined in [23] as:

   > " The unlinkability of two or more items of interest IOI (e.g., subjects, messages, actions,...) means that within the system [...] the attacker cannot sufficiently distinguish whether these IOIs are related or not".

   For example, hide the link between two anonymous posts sent by the same person or the relationship between two users in a social network.
4. **Unobservability** refers to hiding the relationship between two activities of the same user [23]. In other words, an attacker can't link an activity observed to any user.
5. **Data protection** is defined in NIST (National Institute of Standards and Technology) Special Publication 800-33 [24] as:

   > "The requirement that private or confidential information is not disclosed to unauthorized individuals".

   It is not concerned only with protecting the stored data, but also with protecting data in transit. It can be achieved using encryption.
6. **Data access** defines the capacity of getting information about a user of the service. The fetching of information can be performed by the followers, the service users, the internet users, the service provider, etc. For example, who can access the data shared by the user? Who's following who? Who's interested in what?
7. **Users' authorizations** are the access policies defined by the users of the system in order to authorize or deny the other users or the service provider the access to their data.
8. **System's authorization** refers to the access rights that the service provider has in order to access the users' data stored or in transit and the content of the data.
9. **Censorship-resistance** property prevents the system provider from denying access to a particular piece of information (file, resource...) and ensures that the information is accessible to all authorized users anytime and anywhere.

3. OSNs: The Current Picture

We have selected 24 OSNs for our survey, the complete list is provided in table 1. The systems differ in their design choice, the functionalities they provide, and the security and privacy models. We describe each system based on the properties explained in section 2. Some surveyed systems don't provide information on all the analyzed properties and so, only their public features are considered in the comparison.
We have classified the different systems into two classes: (1) deployed systems that are in service and operational, and (2) not deployed systems that are proofs of concepts or proposals in the literature.

3.1 Deployed Online Social Systems

In this section, we present a brief description of the 12 OSNs in our survey that are deployed and operational. The deployed systems provide their users with privacy settings to tune the level of privacy desired, and with privacy policies that disclose what is the data gathered and how it is used, managed and disclosed depending on the applicable laws. However, most of the deployed systems retain the right to modify the terms of the privacy policies at any time. In addition, they generate their revenues by processing, analyzing, and aggregating data for advertisement purposes.

3.1.1 Facebook

It was created in one of the dorms of Harvard University in 2004 [25]. Facebook provides social networking services to its users where they can share their daily life with friends and connections. Currently, Facebook has 1.86 billion monthly

Table 1. List of Online Social Networks

OSN Proposal/ System	Year of publication	OSN Proposal/ System	Year of publication
Deployed		**Not Deployed**	
Facebook	2004	PeerSon	2009
Twitter	2006	Safebook	2009
Jaiku	2006	FETHR	2009
Tumblr	2007	Megaphone	2010
Plurk	2008	LifeSocial.Kom	2010
Pump.io	2008	Cuckoo	2010
Diaspora	2010	Vis-à-Vis	2011
Twitsper	2013	Garlanet	2011
Twister	2013	HummingBird	2012
trsst	2013	DECENT	2012
http://gab.ai	2014	Cachet	2012
GNU Social	2014	Twitterize	2013

active users as of December 31, 2016, which approximately 85.2% are outside the US and Canada [2].

Facebook uses a centralized architecture with MySQL database infrastructure and Global Transaction ID with MySQL semi-synchronous replication [26, 27], where the availability of the services depends on the single authority of Facebook.

Facebook gives their users the possibility to create accounts, add, accept or decline friendship requests. Users can easily create their profiles providing some personal information like the full name, phone number or email address, etc. Users can post text messages, files, videos, etc. in their wall, and they can reshare or comment others' posts. They can mention other friends and they can post directly on their friends' walls, provided that they are authorized to do so. Users can also follow their interests by following or creating Facebook pages. Facebook gives its users the possibility to privately chat using instant messaging. Also, it displays recommendation based on the location and interests of users and it gives the possibility to the users to search for a user, a page or a group. Facebook's profile is by default public and anyone on the Internet can access it and see what is shared and the relationship between the users.

The provider assures that all communications between servers and clients are encrypted using HTTPS secure channels. Recently, users have the option to encrypt and authenticate their communications in instant messaging "Facebook Messenger" using AES_CBC and HMAC_SHA256 [28, 29].

Facebook doesn't provide anonymity or pseudonymity. In fact, recently, it has implemented a real-name policy for user profiles and the policy reads: "You will not provide any false personal information on Facebook" [30]. Facebook provides basic privacy settings for users to choose from, where they can restrict their profile to be private or public and they can choose who can access their profile and see their posts. But, since the architecture of Facebook is centralized and not encrypted, the right to access all information stored in the database stay in the hand of the provider, and all deleted contents persist in the backup copies for a period of time, making the act of censorship easier. In fact, in the privacy policies, Facebook states that they retain the right to disable an account if they see it fit [30].

3.1.2 Twitter

It is a microblogging service provider, created in 2006 [31]. Twitter allows users to post, retweet, and comment on short 280-character messages called "tweets" [32]. It has more than 328 million monthly active users from which 79% of the accounts are from outside the U.S [1, 33].

Twitter adopts the centralized architecture and it has built a next generation distributed database to match their need for availability, scalability and real time interactions [34]. However, Twitter had experienced many outages concerning availability, as it happened in 2014 after Ellen DeGeneres' tweeted an Oscar selfie [35].

Twitter allows users to create profiles by providing personal information like full name and phone number. Most of the information provided in the profile are always public like biography, location, and picture. Users can post photos, videos, and location information. Also, they can mention other users. In this case, the mentioned users will see the message in their timeline although they do not follow the sender. Users can also search messages related to a certain topic, and they can look for and subscribe to other users' tweets. People may also find other users through third-party services that have been integrated with Twitter. The Twitter's interface displays a list of trending topics on the sidebar along with recommended contents or potential followees.

Twitter uses Transport Layer Security (TLS, formerly SSL) to secure the communications between the clients and the servers, and it provides optional verified Twitter account where the user can submit a request to authenticate the identity of the

person or company that owns the account [36].
Twitter wasn't built with the protection of privacy and anonymity of users in mind. The profile, tweets and list of followers are public by default and accessible to all the Internet. But, users can restrict message delivery to just their followers or to just one follower in the case of direct messages. However, Twitter retain the right to access the data stored and analyze its contents to ban abusive and offensive hashtags or users [37].

3.1.3 Jaiku

It was developed in 2006 as one of the first competitors of Twitter offering microblogging services. Jaiku was acquired by Google in 2007 [38]. The number of users is not known as Jaiku was shut down in 2012 [39]. Jaiku was based on a centralized architecture where centralized databases were responsible of storing profiles and data of users.
Jaiku allowed users to create profiles, to send and comment on posts, to mention other users and to tag interests. The posts were limited to only 100 characters. Jaiku released an API that allows programmers to integrate Jaiku services in their software. It offered also Lifestream, a feedstream service to share online activities [40].
The profiles and posts were by default public and visible to everyone, but the users had the option to make their profiles and posts private to only their subscribers.

3.1.4 Tumblr

It is a popular online social networking website [41] with more than 345 million active users by April 2017 [3]. Tumblr is operational since 2007 and owned by "Yahoo!" since 2013 [42].
The platform uses centralized architecture with Redis, HBase and MySQL databases [43] and Multi-source Replication from MariaDB [44] to protect the availability of their services.
Through the Tumblr dashboard, users are able to post texts, images, video, quotes, or links to their blogs, to comment or share others' posts, to tag interests and to mention other users. The profiles in Tumblr are by default visible to all Internet. Since 2014, Tumblr released a new update that allows the users to hide their blog from the web and be only viewed for the users of Tumblr.com [45]. Tumblr gives recommendations of possible friends or interests to its users based on their previous activities. Through the search box implemented in the dashboard, the users can use the email address of the blogger to find a new blog, provided that the blog author has enabled that setting. Tumblr uses TLS to secure the communications between the clients and the servers.
Tumblr's users can restrict the accessibility of their blogs. The users can hide their Tumblr blogs from public search. But even in that case, the profile and all the posts shared on the blog are visible to the other Tumblr users even if they don't figure in the followers' list. Tumblr collects personal information such as name, age, email address, location, and financial information, like credit card number, type, expiration date or other financial information as stated in their privacy policy [45].

3.1.5 Plurk

It is an OSN that provides microblogging services, launched in 2008 [46]. It allows its users to send short messages (up to 210 text characters in length), links, videos, and photos. It's estimated that Plurk has more than 1 million daily active users of which 71% are from Taiwan [47]. Plurk uses a centralized architecture where the data (users' profiles, messages, IP address) are stored in MySQL databases.
Plurk allows users to create profiles using personal information such as full name, email address and birthdate. To add friends, users send friendship requests to establish a mutual relationship, but they can also follow others without their permission. The users can send messages to groups or to individuals using instant messaging. Users can reply, reshare posts, or mention other users and tag their interests. Plurk also provides a mechanism to recommend or search for other users or interests. All communications use HTTPS secure channels. Plurk gives its users to choose to allow everyone to see their profile and timeline as they can make the profile and posts visible only to friends. They have the possibility to send anonymous posts, but all data are stored at the level of centralized databases making the service susceptible to censorship.

3.1.6 Pump.io

It is an open source censorship-resistant social network that provides microblogging service [48]. It was known previously as Identica.ca [49] but since 2013, Identica.ca has stopped accepting new registration and migrated to pump.io.
Pump.io uses a distributed architecture with a federation of servers. Users can choose where to sign up, and save their data. Users might also build their own server and host the services of the social network.
Users of Pump.io can send messages, comment or share others' messages, tag interests and search interests and users. By default, a post is only visible to the users' followers. The users can make the post visible to everyone on the Internet by including 'Public' in the 'To:' box. The communication between servers is secured using TLS certificates.
Pump.io has the ability to hide the profiles and data from the general public in case the users opt to create their own servers. Otherwise, the administrators of the servers have read and write rights to access the data stored on the servers.

3.1.7 Diaspora

It is the first federated, user-owned OSN that is deployed and operational since 2010 [15]. Diaspora has more than 1 million active accounts and it grows continually [50]. Diaspora is based on the free Diaspora software [51].
Diaspora has a federated architecture, which allows users to create their own server/pod and host their accounts. The users can choose also to create their profiles on an existing pod. They can choose a pod based on the physical location, the frequency of updating software version, the domain name or the ratings of the pod. Users can join a pod that is open as they

can join a closed pod upon receiving an invitation. To ensure the availability of the data, the Diaspora network distributes data replicas to multiple pods.
User's profiles have a public part (name, interests, and photo), and a private one with detailed information (biography, location, gender, and birthday), which is only visible to people which users authorize. In Diaspora, it is possible to follow another user's public posts without the mutual following requests required in some other social networks. Diaspora does not show the friends' lists to other users and it has two ways to follow a user:

1. If the follower is located in the same pod as the followee, s/he can use her/his pod's search feature to connect to the followee.
2. If the followee is located on a different server; the follower must know the entire Diaspora handle (ID) to find the followee.

Diaspora offers two options to publish posts: either (1) publicly where any logged-in user can comment on, reshare, and like the public posts, or (2) privately where only followers placed in an authorized group can comment on and like the private posts. Private posts are not resharable. Posts in Diaspora can include mentions and intersts. Diaspora offers also instant messaging services called conversations where users can send private messages.
Diaspora uses Pretty Good Privacy (PGP) where a unique public/private key pair and an ID called guid are assigned to every user created on a pod. The pod is the one in charge of encrypting and decrypting requests before passed to users.
Diaspora focuses on three aspects to offer to its users: (1) censorship resistance, (2) privacy and control of data, and (3) the freedom to choose what and with whom to share posts. The administrators of pods have read and write access rights to the unencrypted data stored on their pods [52].

3.1.8 Twitsper

Singh et al. introduced Twitsper [14], a wrapper over Twitter that provides privacy controls to the users of Twitter. Twitsper was built on Android in 2013 to protect Twitter users' browsing habits and routines and at the same time to preserve the commercial interests of Twitter.
To be compatible with Twitter, Twitsper uses the same centralized architecture adopted in Twitter. In other words, users can preserve their privacy while sharing updates on Twitter, without migrating to a new application or a new OSN. The private messages in Twitsper are called whispers. Twitsper considers one to one messages technique to send whispers to a group. The users' profiles and whispers are stored on Twitter's servers while the Twitsper's servers store, in MySQL databases, the mapping between the hashed message IDs and the list of users involved in the chat group. The availability of Twitsper relies on both the availability of Twitter services and the Twitsper's server. In case the Twitsper's server is offline, the users can continue using Twitter's services normally without the privacy option. The system puts its trust on Twitter servers not to leak the user's private information.
In Twitsper, the users continue to have the same functionalities that Twitter offers: create profiles (public or private profiles), follow interests, post, comment, share tweets with one or a group of followers, search for content or users and get recommendations. Besides, it offers the whispers to its users.
The Twitsper system uses TLS certificates to validate the server's authenticity. To hide the identities of the users involved in a whisper from Twistper's servers, the list ID is encrypted with a group key using AES. The recipients of the whisper derive the group key from each message. So, even if the group key is exposed at any moment of the conversation, it doesn't reveal the old nor the future messages sent to the group.
The groups in Twitsper are created and administrated by the users at the level of Twitter. To reply to a whisper, the user replies only to an intersection between the members of the recipients of the original message and her/his followers, then sends a direct message as a reply to all the users in the intersection. In doing so, the user has restricted the visibility of the reply to only the followers s/he approved of.

3.1.9 Twister

It is an open and free platform, operational since 2013 with 10000 registered users up to date [53]. It offers microblogging services to its users. Twister has a decentralized architecture composed of three overlay networks: (1) a user identity creation and authentication network based on the Bitcoin protocol, (2) a Distributed Hash Table (DHT) overlay network used for resource (i.e. avatar, profile) storage and retrieval, and (3) a collection of disjoint groups of followers network used for notification delivery [54]. The messages of the users are stored in two networks: (1) a short-lived value stored in DHT network and (2) an archive file similar to BitTorrent network.
Twister uses blockchain mechanism to create the users' profiles and to guarantee their uniqueness. To propagate user's posts, Twister uses BitTorrent, and anyone who joins a user's torrent can follow the posts. Followees are not notified and do not need to authorize the followers. The users of Twister can send messages to read-only users or to a group of followers. They can send also direct messages (DM), provided that the recipient is a follower of the sender. The followers can also reply to a post, tag a topic or mention a user in a post, but they can't republish posts of other users. The system provides its users with the option to search for arbitrary words, but not with a recommender.
Unlike other deployed microblogging systems, the users in Twister can't be identified if they are online or what posts they are reading. Twister ensures the anonymity of the senders and prevents to identify the users by forwarding the posts using a number of intermediate nodes before reaching the final destination.
Twister is designed to protect the freedom of speech and

the anonymity of user's activities in the platform, also it's censorship-resistant since there is no central authority to administrate the system. Twister uses ECIS (Elliptic Curve Integrated Encryption Scheme) to end-to-end encrypt the data of users and to digitally sign messages ensuring the authenticity and the integrity of users [55].

3.1.10 Trsst

It is a Twitter-like microblogging system [56], deployed as an alpha test in 2013 [57]. It adds encryption, anonymization and censorship resistance to protect the privacy of its users.
Trsst uses a distributed network where a federation of servers agrees to store and propagate the feeds to users. Trsst's users have the possibility to create standalone client nodes and store their profiles and feeds or they can contract with an existing server, known as the home server to store the keystore, the feeds and the attachments. A copy of the stored data is sent to Trsst hub (home.trsst.com/feed) for replication [58].
To create one or more accounts, a user first creates and encrypts a keystore with a password. This latter is used to access and modify the keystore. The user then generates a keypair, and stores it in the keystore. The users of Trsst may optionally attach personal information to their account, such as name, nickname, image, etc.; the users also have the choice to remain anonymous. The users can send, comment and share texts, images, videos, or files with everyone or with only one person in case of the instant messaging mode. Trsst offers the possibility to search for users knowing their IDs (the users' public key). Also, users can follow and mention other users or tag interests in their posts. Trsst can recommend a list of other users to follow.
Trsst uses encryption of messages to protect the security of contents from censorship. In fact, Trsst uses both public-key and symmetric cryptography. Trsst uses a crypto-currency system such as Bitcoin to generate the keypair. The account's private key is kept in the keystore. To encrypt a message, the user generates a new AES 256 key and uses it to encrypt the message, and then s/he encrypts the generated key using ECDH (Elliptic curve Diffie–Hellman) and appends it to the encrypted message [59]. The result is encrypted with the intended recipient's public key. All client-to-server and server-to-server communications are conducted over HTTPS channels and all public posts are digitally signed.
Even if Trsst promotes the protection of the privacy of users, it is still suffering some aspect that might endanger the security and the privacy of users. In fact, Trsst users' profiles are public to anyone who knows their IDs and also a user can start a conversation with others without following them. Moreover, the list of followers is available to the public, and anyone on Internet that knows the user's ID can check his/her posts unless the post is private.

3.1.11 Gab.ai

It was launched in August 2016 [60] and has 215,000 active users [61]. Gab offers microblogging services that allows users to post, reply, and republish short messages called gabs. Gab comes in two versions: the free and limited Gab or GabPro. GabPro is a paid and more elaborated version that allows users to create lists, use private group chats, and to go live [62].
Currently, Gab uses centralized architecture to store and replicate data on servers. However, the creator of the system have announced that they will change the architecture in the near future to a decentralized architecture in order to build a true censorship-resistant and community-powered system [61, 63].
Gab has become open to the public recently as it was limited to join by invitation before. The users can create a profile using a username, password, and an email and they can choose to make their profiles public or private. Once the account is created, the users can add new followers and they can send, quote a post, tag an interest, or mention another user. Gab enables its users to share up to 300 characters in one gab. The system's dashboard comes with a search box in order to search for other users and interests, and it recommends potential friends and hot topics. The messages sent by users and the lists of followers and followees are public and visible to any user of Gab. Traffic between clients and servers is encrypted using TLS to secure the traffic between the clients and the servers.
Gab was built on the idea of providing freedom of speech and thought. But, Gab service retains the right to administer the users' accounts and store their data and messages in the databases of the platform. In fact, it banned the first Gab user in January 2017 [64].

3.1.12 GNU Social

It is an open source program offering microblogging services [65]. GNU Social was developed for the first time in 2010 and was known under the name of StatusNet project. GNU Social offers similar functionalities as Twitter, but in an open and collaborative environment where the users are in control of their data and profiles.
GNU Social uses a distributed microblogging platform and it has 301 online and active servers to supply thousands of users [66]. GNU Social is composed of multiple instances, the current number of running instance is about 50 instances like Quitter.es, gnusocial.de, loadaverage.org. The instances are independent and they communicate between each other using OStatus standard [67].
GNU Social's users can create profiles using a nickname, email address, and password. They can choose to create an account in any instance and they can communicate, follow and be followed by users from other instances. They can also choose to keep their profile visible and searchable to all Internet users as they can limit the access to only the users of GNU Social. They have the right to choose who can follow them and who can read their posts. The users can send texts, files, images, videos, and audio to all GNU Social users, or to private groups, or only to one individual as a direct message. The users can share and comment on a post and they follow an interest.
GNU Social focuses on the availability and the censorship-

resistance. The fact that there is no central unit that can bring down the whole network or censor the content of messages reinforces the GNU Social's position in protecting the freedom of users. Also, secure channels between users and servers and between servers are used to protect the confidentiality of messages.
However, GNU Social suffers from privacy issues. Actually, the activity of users is public on their timeline and the lists of followers are disclosed to anyone even the unregistered users. GNU Social also lacks controls to protect the integrity and the confidentiality of users and posts from the administrators of the instances, considering that data are stored in clear in the databases. In fact, the administrators can have access to the users' posts, they can read or delete them, and they can even ban a user from using the services of GNU Social.

3.2 Not Deployed Online Social Systems

In this section, we present a brief description of 12 OSN solutions that have been proposed in the literature. These systems are proposed as an alternative way of how to effectively address security and privacy protection in OSNs.

3.2.1 PeerSon

It is a proposal for an OSN using a decentralized architecture that provides encryption and access controls coupled with a peer-to-peer (p2p) approach to replace the centralized authority of classical OSNs [68].
In the proposed version of PeerSon, the developers suggested using open DHT for the lookup service to store the data, and to replicate the social links and digital personal spaces (i.e. timeline, posts) in other nodes.
PeerSon proposed to use e-mail addresses as unique identifiers of the users. In order to prevent a malicious DHT-node from collecting e-mail addresses, PeerSon computes a user ID based on the hash of the e-mail address. The users can look for a specific user to follow using the lookup service directly to get all necessary information. They can post and reply on messages and the can also control who reads and replies on their messages. PeerSon uses public key cryptography to encrypt the messages with the target peer's public key, hence the messages are only accessible to those who have the right keys.

3.2.2 Safebook

It proposes a distributed OSN to protect the privacy and the availability of the messages. Cutillo et al [69, 70, 71] proposed a three-tier architecture for Safebook:

1. The first tier, called Matryoshkas, handles communication privacy, data storage, and availability of data. A Matryoshka has a core user, surrounded by her/his friends (first shell), friends-of-friends (second shell), etc. All published posts are stored on the user's machine and replicated to a mirror group created by the user based on her/his friendship with other users from the first shell.
2. The second tier is a peer-to-peer (P2P) overlay that provides the application services (e.g., lookup service, identity management service, etc...)
3. The third tier is a Trusted Identification Service (TIS) that provides each user with a unique identifier and public/private keys.

To join Safebook, a user needs an invitation from an already registered user. The new user provides her/his identity set and a proof of owning it, and generates a public/private key pair. Then the TIS computes a unique identifier and generates a certificate associating the public key of the user with the identifier. Once the new user is registered in the system, s/he can start the process of creating her/his Matryoshka by sending friendship requests. Each new friend is associated with a trust level with appropriate privileges to who can access the user's profile and read her/his posts. The users in Safebook are notified about a new friend request and they can accept it or discard it. When the request is accepted, the two friends exchange their respective certificates to start communicating. Safebook doesn't provide the options of mentioning other users or tagging their interests, but the users can share text messages publicly if the post is tagged public or only with a group of chosen friends if the post is tagged private. The users can also comment or republish messages.
The developers of Safebook were concerned with building a system that provides end-to-end confidentiality, authentication, access control, censorship resistance, data integrity, and data availability. Safebook categorizes data into three types: (1) private data (unpublished), (2) published and encrypted data, and (3) published data without encryption. All exchanged messages are encrypted using the receiver's pseudonym public key and signed with the sender's pseudonym private key. The communication tracking in Safebook is not possible since it was built with the concept of Matryoshka. In other words, the malicious node needs to be the first hop for all requests going from and to a node in the Matryoshka to intercept the communications. Also the mapping between the user's identifier and the pseudonym is only known to the TIS and the direct first shell of friends.

3.2.3 FETHR

Sandler et al. [72] proposed a new infrastructure to integrate microblogging services called FETHR (Featherweight Entangled Timelines over HTTP Requests). FETHR enables users to communicate with each other on top of HTTP with messages of more than 140-byte payload.
FETHR proposes a decentralized architecture where users' data are stored locally on each peer's machine and new messages are gossiped to the followers using a lightweight HTTP-based protocol.
Each user has a canonical URL that serves as a unique ID. This URL contains the user's profile with the personal information and the messages published. The canonical URL is public and searchable by any other user. Followers can subscribe to another user's update simply using HTTP GET and POST

messages. FETHR uses a gossip-based update propagation technique where the message's publisher pushes the update to a subset of the followers, who in turn push the message to the rest of the network. The gossip technique plays a role in the distribution of messages and also in the protection of the data against suppression.
The objectives of FETHR don't include privacy preservation controls, it is concerned more about the availability, the authenticity, the integrity, and the completeness of messages. Also, FETHR uses some cryptographic measures such as hash chaining and digital signature to preserve integrity. The decentralized architecture of FETHR ensures that the system is censorship resistant and not reliable on any single service.

3.2.4 Megaphone

It is a proposal of a multicast microblogging system based on a peer-to-peer network [73]. Megaphone organizes the social graph of users in multicast trees where a "poster" node is the root of the tree, and a "follower" is a child node.
The storage of data is performed at the level of the roots and replicated in child nodes. With the decentralized architecture, Megaphone insures that the system is censorship resistant considering that there is no central authority responsible for administrating the service.
The poster creates the tree, manages the join requests and the list of followers, stores the public keys of child nodes, and sends messages to all nodes in the tree. The poster has the right to accept or discard the new join request. A follower can post a response to a message from the poster, and optionally encrypts and signs it.
Megaphone uses public key cryptography based on RSA. The poster generates session keys to encrypt the messages. The session key is cached by all nodes of the multicast tree, and readable only by the nodes that have registered a public key with the poster. The poster might add a serial number to detect lost messages.
Using the multicast architecture, Megaphone protects the confidentiality, the integrity, and the availability of data. Megaphone protects also the anonymity of users since the IDs are not based on any piece of information related to the users' real identities, but rather on their public keys. However, the followers inside the circle of the multicast trees can know the source of the posts and who is currently following the poster.

3.2.5 LifeSocial.KOM

It is a decentralized OSN based on peer-to-peer network [74]. It was built to offer the social functionalities of an OSN, with a fault-tolerant and data storage efficiency.
All personal information and shared messages in LifeSocial.KOM are stored in the peers. It provides data availability using the replication mechanism offered by PAST [75]. PAST is an Internet-based, peer-to-peer global storage utility that aims to provide strong persistence, high availability, scalability and security.
The users of LifeSocial.KOM can create profiles, manage the followers' lists, create, join and manage groups, follow interests, share text and photos, search for people with common interest, browse through pictures of friends and interesting people, and live chat or multi chat with their friends. The profiles and the posts of the users are only visible to the friends.
LifeSocial.KOM focuses on providing confidentiality, availability and access controls to its users. It uses public key cryptography for authentication (the public key is used as a unique ID of the users) and the symmetric cryptographic key is used for encryption. For the access control, LifeSocial.KOM suggests a user-based access control to access the system where users can control who can read and access their data. [76]. Leveraging the decentralized architecture of P2P networks, LifeSocial.KOM protects against censorship since no central authority is responsible for providing the service.

3.2.6 Cuckoo

It was proposed in 2010 by Xu et al [77, 78]. Cukoo was one of the earliest microblogging systems proposed in the literature that leveraged the decentralized architecture of peer-to-peer networks.
The architecture of Cuckoo is hybrid, meaning it's composed of a small servers base, named server cloud, and client peers. The server cloud is used for storing resources like users' profiles and served also as a backup for replication to guarantee the availability. The client peers are served as an overlay of the messages.
The profiles and messages sent by the users are public. Anyone can search information about any other user. Besides the known categories of relationship (followers and followees), Cuckoo gives its users the possibility to organize their social relationship into friends (the two users reciprocate the social link between them) and neighbors (users who serves as an overlay to disseminate messages based on gossip protocol).
Optionally, Cuckoo uses asymmetric key cryptography to encrypt and to sign the messages. The public key is stored on the server cloud while the private key is kept secret in the client peer's machine. The users can obtain the public keys of the followers either out of band during the following process or from the server cloud.
Cuckoo focuses on providing a microblogging system that is scalable, reliable and censorship resistant. In fact, Cuckoo protects the users only from censorship since the server cloud is used for storage of profiles and doesn't intervene in the message exchange between peers. However, Cuckoo don't take the privacy protection of the users into consideration.

3.2.7 Vis-à-Vis

It is a decentralized framework for OSNs based on the privacy-preserving technique of a Virtual Individual Server (VIS) [79, 80]. VIS is a highly available virtual machine running in a paid compute utility, like Amazon EC2, which don't have any claims over the contents stored in the machines. VIS are used to store the users' personal data and posts.
The communication between users is conducted in groups where they can share posts and follow interests, but they can't comment or republish the posts. Each group of users consists

of an administrator who create and manage the group, the members (other users) and the mapping of members in geographic regions. Each member maintains an attribute within the group such as the relationship with the administrator or an interest in a particular topic. Users also have the option to search for a group or a user in a particular region, but the system doesn't provide any recommendation of available groups or users.
Vis-à-Vis uses public-private key encryption, where users are defined by a self-signed key pair. The public key is used in the encryption of messages and the private key is stored securely in the VIS and it is used for digital signature and decryption of encrypted messages. The public key of a user and the corresponding IP address of the VIS are distributed out of band. Vis-à-Vis is concerned mainly by the AIC triad (availability, integrity, and confidentiality) of security more than privacy. In fact, VIS administrators can access to all users personal data stored on their machines, but the intermediate computers can only access the ciphered data and some other control data (users' ID and timestamp). Thus, VIS owners need to manage securely their machines, keep them up-to-date and implement the appropriate access controls policies.

3.2.8 Garlanet

It is a privacy-preserving microblogging system developed at Universitat Oberta de Catalunya (UOC) [81]. It is a collaborative system where the registered users are voluntarily contributing to the computational resources.
Garlanet uses a hybrid architecture composed of a directory service and clients' peers. The directory service is used for lookup services and location data. Users' data are hosted at any resources provided by any users of the system. To ensure the availability of the service and data, Garlanet replicates the data of users on different machines.
Garlanet offers its users with the possibility to stay connected with their followers and to express themselves in a censorship free system. Users can share their activities and interests with their followers and they can also follow other users of the system. In Garlanet, the following process is one-sided and is conducted out of band. Users can access only the public information (name, username, and photo) of another user and they can't access the private information provided in the other users' profiles even if they are following them.
Garlanet is a community-owned OSN where no central authority controls the system. It adds built-in privacy mechanisms to guarantee that only the sender and the intended receivers are able to access the information exchanged. These capabilities can protect the end users from the malicious utilization of personal information and from public exposure of sensitive data, and, they guarantee the free exchange of information.
Garlanet protects the confidentiality of sensitive data and guarantees the desired level of anonymity of the users. Each user in Garlanet uses RSA to generate two public keys: (1) one to cipher the storage and (2) the other key is used to decipher the user's messages. The friendship relation between users is not revealed to anyone and the users only have the list of the contacts who they are following. The data are distributed in different repositories so an attacker cannot get information by correlating all the actions that a user performs. Also, the intermediate computers only see the ciphered data and some control data such as a pseudonym ID or a timestamp.

3.2.9 Hummingbird

It is a microblogging system that imitates Twitter's functionalities while adding privacy-preserving techniques to protect the personal data of users [13, 82]. Hummingbird imitates Twitter using centralized architecture where the Hummingbird Server (HS) handles all the operations of the user's registration and tweets delivery to followers.
Hummingbird introduces the new concept of "follow-by-topic", where users can decide to follow other users on specific hashtags of interest. It also allows users to conceal their interests by following arbitrary hashtags. A follower issues a request to follow a user on a specific hashtag and the follow requests are subject to approval. To preserve privacy, Hummingbird doesn't allow users to reply to a post or reshare it with other followers. The users' profiles are visible to all other Hummingbird users.
Hummingbird uses several cryptographic protocols like Oblivious PRF (OPRF) technique and Blind- RSA for signature. The users are responsible for generating their own keys and distribute them out of band. The keys are stored in HFE (Hummingbird Firefox extension). The proposed architecture does not handle revocation of the following requests.
Hummingbird is concerned mainly about providing confidentiality and authorization. It adds encryption of tweets to provide confidentiality and access lists for users in order to choose who can access their messages. The post are hidden from the server and all non-followers and the access to them is restricted only to the authorized followers. However, the Hummingbird server has the access to users' accounts, encrypted messages, and to the following requests. It can build a full graph of tweeter-follower relations. In addition, the server can learn whether two followers are subscribed to the same hashtag of a given user and it can learn whenever two posts by the same user carry the same hashtag.

3.2.10 DECENT

It is a proposed project for OSNs that suggests to use a fully decentralized architecture and store user data in a Distributed Hash Table (DHT) overlay [83]. Each write operation in the DHT storage requires a prior authorization. This authorization doesn't reveal the social graph of a user. To protect the objects stored in malicious nodes from vandalism and deletion, DECENT maintains several replicas of an object of a node among its neighbor set, providing high availability to users.
A profile in DECENT contains references to biographic information, the list of contacts, a wall, and a photo albums. The users can search for a profile using the wall reference. The users of DECENT can post messages, links, photos or videos, add a comment, refer to an existing object, and mention another user from their list of contacts. Relationships in

DECENT are asymmetric and the users affect levels of trust to their followers. In other words, the level of trust affected to a user might not be reciprocated. For example, user A can add user B to her list of contact just as an acquaintance relationship, while user B can label his relationship with A as friendship.
DECENT provides the confidentiality, the integrity, the availability of the message's content, and the privacy of user relationships. DECENT uses AES for symmetric encryption, DSA for signatures, and RSA to encrypt the write policy signature key. DECENT uses also an extended version of EASiER [84]. EASiER is a fine-grained access control architecture for OSNs that uses Attribute-Based Encryption (AB encryption) [85]. The users retain the master secret key hidden and the keys are exchanged out of band.
When creating an object, the sender creates 3 policies related to the object that state who can read, modify/delete, or comment/annotate the content.

3.2.11 Cachet

It is proposed as a performance improvement of DECENT [86]. Cachet maintains the same functionalities and services of DECENT. It uses a decentralized architecture to provide social network services with strong security and privacy of data. Cachet uses also Distributed Hash Table (DHT) overlay network similar to DECENT to store data and replicate it in the selected nodes ensuring high availability of the objects. The data in Cachet are stored in containers that include updates and photos, wall references as well as references of other containers. The containers are protected by encryption.
Cachet uses Attribute-Based Encryption (ABE) scheme [85]. All the keys are exchanged out-of-band. The message is encrypted using a symmetric key which in its turn is encrypted with ABE. Cachet uses the digital signature to insure the integrity of objects. Also, users maintain secure connections with the followers to receive new updates directly as soon as they are released. In this upgraded version, the authorized readers don't have to decrypt all the wall object, but only the most recent updates.

3.2.12 Twitterize

It is a system designed to preserve the privacy of Twitter's users [87]. Twitterize is built to overcome the shortcoming of Twitter in terms of anonymity and confidentiality. It offers the option to send posts anonymously while maintaining the normal Twitter functionality.
Twitterize uses the twitter4j library and it maintains the same centralized architecture of Twitter. Twitterize uses Android SQLite DB to store tweets, cryptographic keys, subscriptions, etc.
To achieve anonymity, Twitterize doesn't require P2P communication but establishes one overlay network per each hashtag to connect the sender and receiver. Each overlay contains forwarders (other Twitter' users who aren't interested in the hashtag) to mix the tweet and forward it to its destination. The overlay network is also used to send subscription requests [88]. Using this architecture, forwarders can't link between the sender and the receiver, they can only control their local view of the message's flow and they can't learn the origin or the destination of the tweet.
Twitterize gives the possibility to its users to create profiles and to customize the behavior of service based on their preferences: the synchronization times and the frequency of tweets to pull during synchronization. To publish interests, the creator of a hashtag x encrypts and hashes it to create a pseudonym Px for the hashtag, then the publisher can annotate Px to tweets without revealing the hashtag. Twitterize encrypts tweets to obtain confidentiality using AES 128bit in CBC mode. The keys are exchanged via an out of band channel (QR code or NFC). Also, the users can generate an optional asymmetric key pair to ensure integrity.

4. Analysis and Evaluation of the Surveyed OSNs

This section provides a comparative classification of the set of OSNs described above with respect to the characteristics detailed in section 2.

4.1 Service provided, architecture and storage

The surveyed OSNs differ in the services provided to their clients, in the architecture, and in the way the data is stored. Table 2 in the appendix summarizes the classification of the systems with respect to the three previous criteria.
Most of the deployed social network sites adopt the centralized architecture using central databases to store the users' data. The main reason for choosing such architecture is because centralized systems are easy to create and to maintain and they offer a better oversight over the data stored. Meanwhile, the decentralized and the distributed systems are more complex and difficult to maintain due to lower level details that should be taken into consideration like resource sharing and communications. However, the single authority provided by the centralized architecture gives the service provider an ownership over the user's data stored in the databases which can be used for monetary gain purposes which presents a threat to the user's privacy. The decentralized and the federated systems benefit from the fault tolerance nature of the decentralized architecture and give the users more autonomy in terms of controlling and choosing where to store their data. When the users opt to host their data on their devices, the system becomes censorship-resistant since no single authority hosts the data and controls the platform. However, in the case of federated systems, the administrators of the pods should ensure the protection of the privacy of the data hosted and the security of the pods. They have to patch, update, and maintain regularly their pods as well as they need to install and manage security tools (firewalls, antivirus, IDS/IPS, ...) in order to prevent data leakage and potential security threats.

4.2 Encryption mechanisms and the key management

All surveyed systems offer cryptography mechanisms to protect the security of the messages and the identity of users. Some systems propose to use asymmetric encryption mechanism providing a key pair (public and private keys) that can be used for confidentiality and integrity. Other systems use symmetric cryptography to ensure the confidentiality of posts. Meanwhile, most of the deployed systems use TLS certificates to ensure secure channels for the communications between servers and clients. The majority of the not deployed systems give the users the ability to generate their cryptographic keys and to manage them. However, the generation and the management of keys in the centralized systems are handled by the providers of the services and the keys are centrally stored. In this case, there is a risk that the system might eavesdrop on the users' messages. Section 3 provides more details of the used encryption mechanisms.
A summary of different aspects of encryption mechanisms in the surveyed OSNs is presented in table 3 of the appendix.

4.3 Functionalities

Preserving the privacy of users comes with a price in terms of the ease of use and the usability level of services provided to the users. The figure 2 compares the distribution of the functionalities between the deployed and the not deployed OSNs.
We can observe that when it comes to the usability of the systems, the deployed OSNs attract their users by offering a richer variety of functionalities and an ease of use. However, not all functionalities are implemented in the proposals. Some proposals don't support services such as commenting or mentioning other users in the posts like the case of Hummingbird, Safebook, Twitterize or Garlanet. The profiles and posts in the deployed systems are open and visible to the public by default while the undeployed ones focus more on limiting the access and the visibility of user's data to other users and to the service provider.
Table 4 of the appendix gives a summary of different functionalities provided by the surveyed OSNs.

4.4 Security Goals

One of the main concerns of OSNs is to protect the users' personal data and prevent the data leakage. Therefore, OSNs need to have robust security features. The security goals of each OSN system as explained in section 2 are summarized in Table 5 of the appendix. Figure 3 compares the two sets of OSNs in terms of security goals.
All the surveyed systems are concerned with confidentiality. They offer encryption of contents with different cryptography mechanisms as explained in the previous section. Moreover, the availability of data depends on the availability of central services in case of centralized architecture or on the replication of data on pods or users' machines in case of distributed and decentralized architectures. Not all the deployed systems are concerned with the integrity of data except Twister and

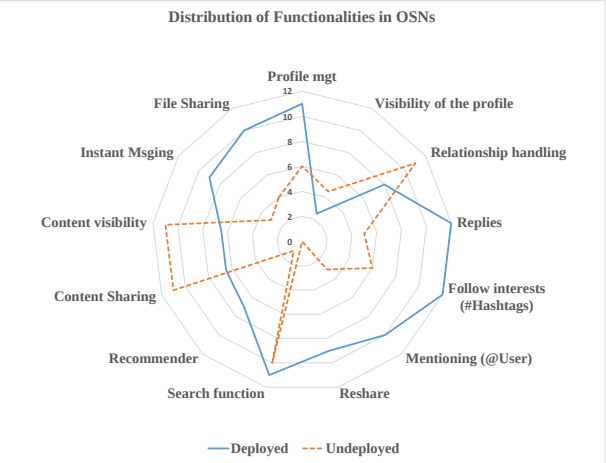

Figure 2. Comparison of deployed and not deployed systems based on functionalities provided.

Trsst, unlike most of the undeployed OSNs that are concerned about protecting the integrity using digital signature.
The mechanisms of the user's identity creation and verification differ from a system to another, but we can observe from figure 3 that all the deployed systems have implemented an identity creation and verification techniques as a way to protect the identity of users unlike the proposals OSNs where they don't give the user means to authentication and verify their credentials but they base their identity creation and verification on public keys or canonical URLs.

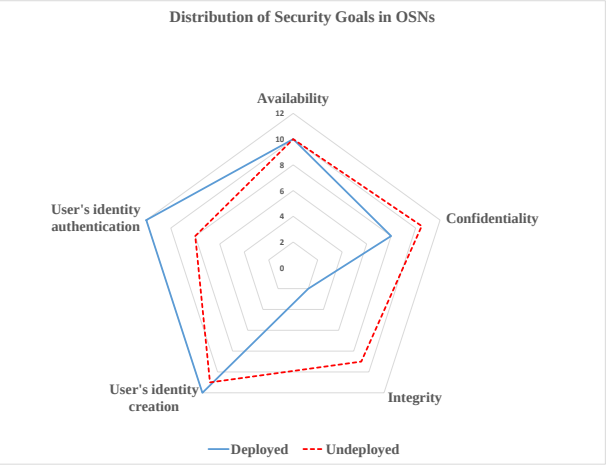

Figure 3. Comparison of deployed and not deployed systems based on security goals.

4.5 Privacy Goals

The surveyed OSNs provide different privacy-preserving mechanisms to control the degree to which the users' profiles and data are visible to the public, to other users, or to the service providers of the OSN.
The undeployed systems are more concerned about meeting

the privacy goals and protecting the anonymity of the users and their data. They provide their users with techniques to hide their identities through pseudonyms like Safebook and Garlanet, and with the possibility to choose where to store the data like Peerson and Safebook. Whereas the deployed systems have implemented privacy settings where the users can limit the access to the data but they don't have the control over it.

Figure 4 represents a graphical representation of the distribution of the privacy goals between the deployed systems and the proposed OSNs. The figure shows that whereas the deployed systems are more concerned about protecting the confidentiality of the messages, the undeployed OSNs protect, in different ways, the anonymity of users, the data access and the access authorizations among other privacy goals. For example, the user ID is never directly revealed to the supporting server/peers or protected under a pseudonym (public key or username provided by the user). Furthermore, the the users of the systems that have adopted centralized architecture don't have any control over their information. The data stored are handled centrally which makes the censorship of information easy. Unlike the decentralized, the distributed or the hybrid systems that give their users the possibility to manage the profiles and posts and choose where to host the data.

Table 6 summarizes the privacy goals featured in the different OSNs.

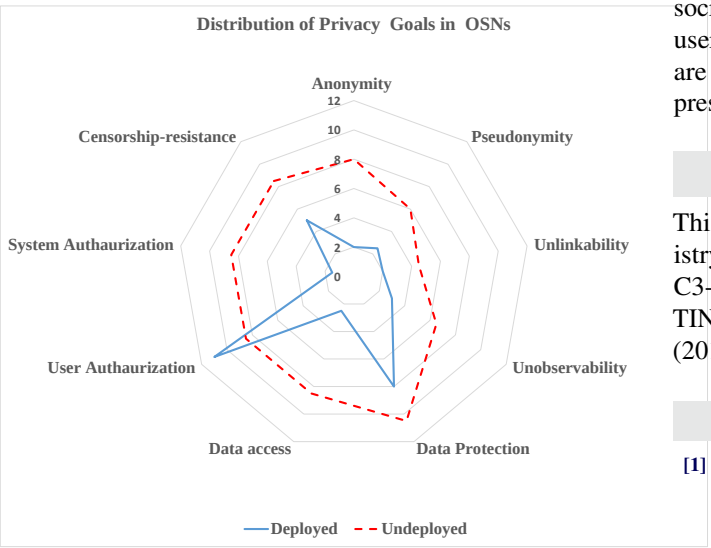

Figure 4. Comparison of deployed and not deployed systems based on privacy goals.

4.6 Usability in OSNs

Usability is concerned with how the system is intuitive and easy to use. When it come to the OSNs, the main functionality is facilitate the interactions between users. Figure 5 and figure 6 show that most of the deployed systems are interested in offering systems that are easy to understand with pleasing functionalities that enable the users to interact with each others which attract more users as shown in figures 7 and 8. However, these functionalities may lead to privacy leakage for OSN users. In fact, the deployed system are less concerned with implementing controls to protect the private life of the users. Unlike the not deployed systems, they worry about protecting the security and the privacy of their users more then the usability of the system.

5. Conclusion

In this survey, we have discussed 24 different OSN systems divided into two categories: (i) 12 systems that are deployed and operational and (ii) 12 systems that are undeployed proposals found in the literature. We have compared the systems based on a set of criteria composed of seven criteria. We have introduced the service provided by the systems, the design architecture, the storage mechanisms, the encryption algorithms, the functionalities provided, the security goals, and the privacy goals implemented in each system. Furthermore, we have presented a comparative evaluation of the surveyed OSN systems based on the security goals, the privacy goals implemented, and the functionalities offered by the system.

This present study gives the scientific community a knowledge base in the field of OSNs to understand more the inner functioning of such systems. It shows also the importance of protecting the privacy and how it can be a challenge especially with the existing trade-off between the usability of a social system and the protection of the private life when even users who are aware of the importance of protecting privacy are willing to endanger their privacy as a price for a digital presence in the virtual world.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (TRA2013-48180-C3-P, TRA2015-71883-REDT, TIN2014-57364-C2-2-R and TIN2015-70054-REDC), FEDER, and the Erasmus+ Program (2016-1-ES01-KA108‐02346).

References

[1] Statista. Number of monthly active twitter users worldwide from 1st quarter 2010 to 1st quarter 2017 (in millions), 2017.

[2] Facebook., 2017.

[3] Statista. Cumulative total of tumblr blogs from may 2011 to april 2017 (in millions), Apr 2017.

[4] Habibul Haque Khondker. Role of the new media in the arab spring. *Globalizations*, 8(5):675–679, 2011.

[5] Matthew Das Sarma. Tweeting 2016: How social media is shaping the presidential election. *Inquiries Journal*, 8(9):1, 2016.

[6] Avery Hartmans. Whatsapp went down all over the world for several hours on wednesday, 2017.

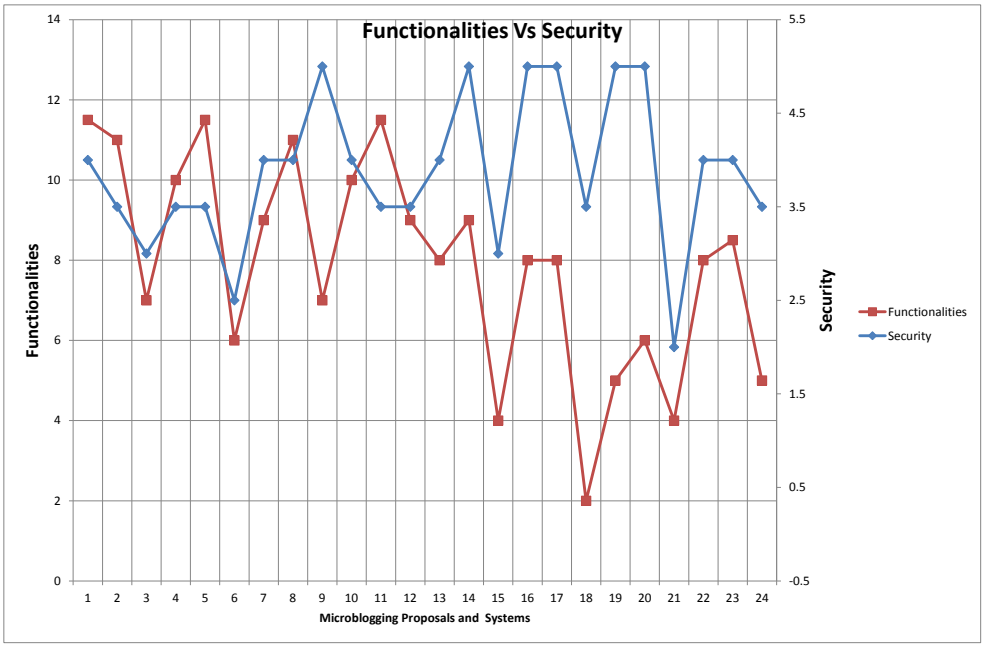

Figure 5. Comparison of security goals and functionalities in OSNs.

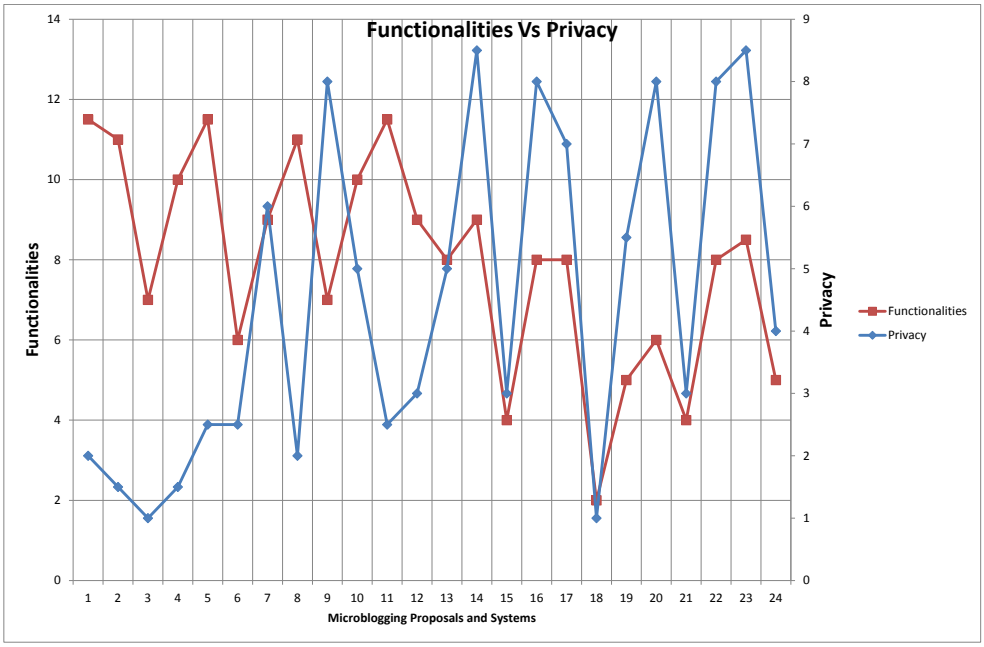

Figure 6. Comparison of privacy goals and functionalities in OSNs.

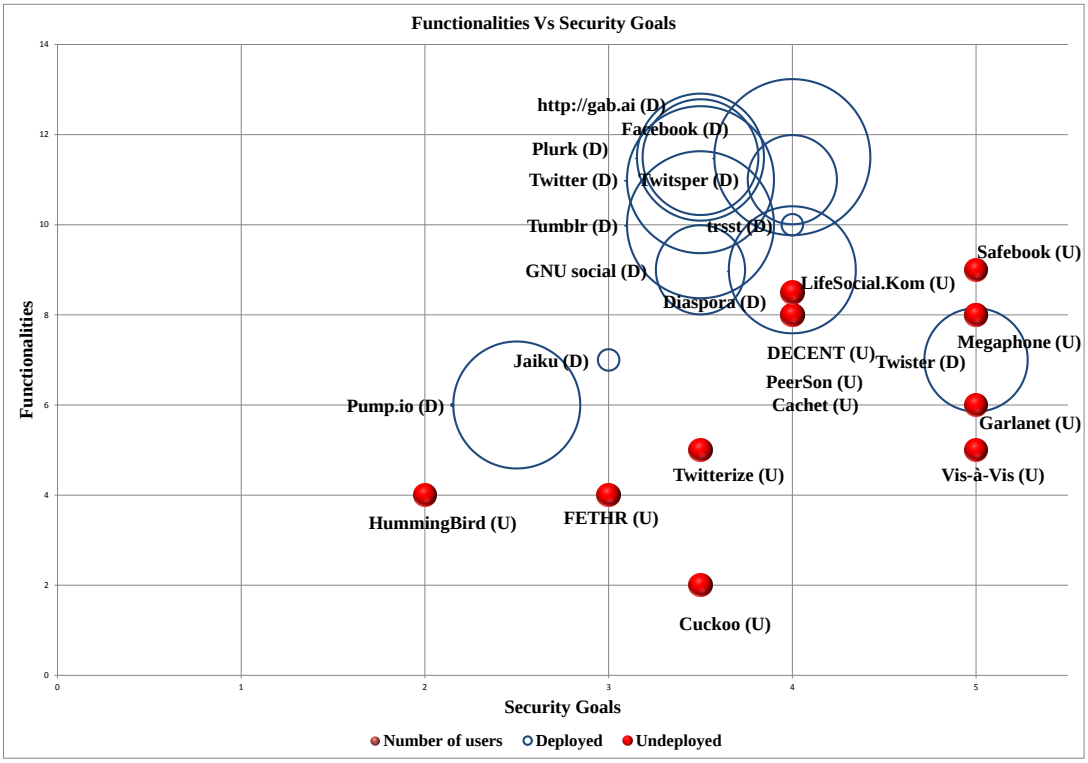

Figure 7. Comparison of OSNs based on security goals, functionalities and the number of users.

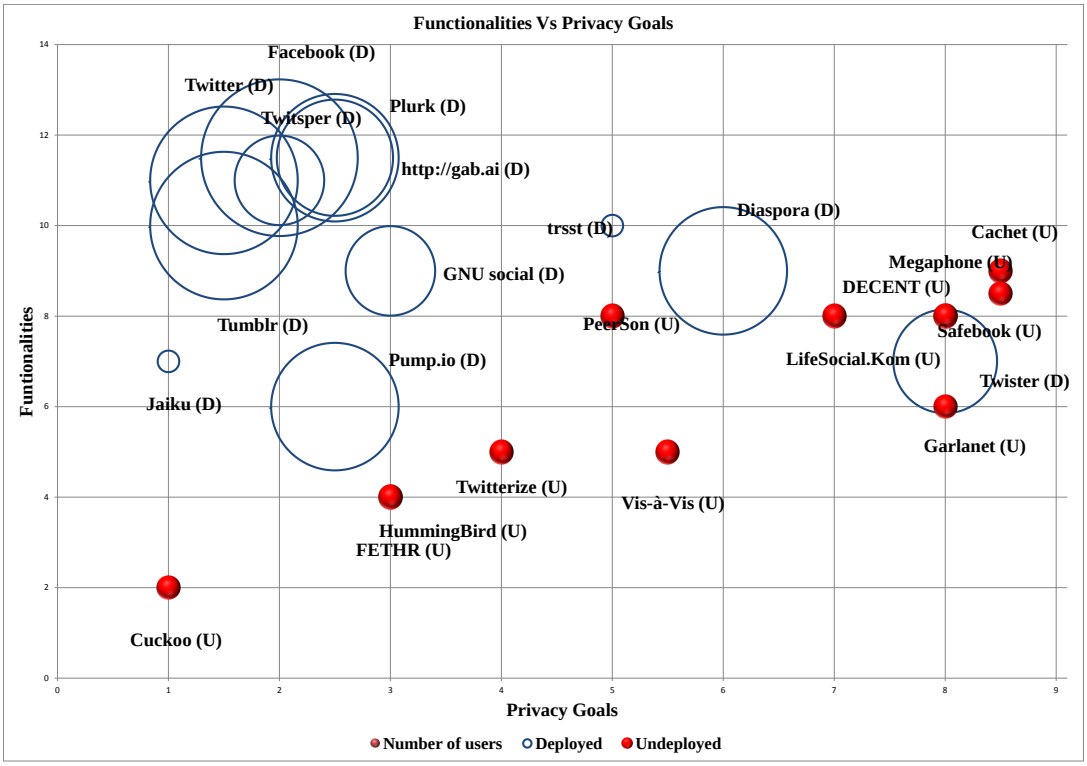

Figure 8. Comparison of OSNs based on privacy goals, functionalities and the number of users.

[7] Michael Fire, Roy Goldschmidt, and Yuval Elovici. Online social networks: Threats and solutions. *IEEE Communications Surveys Tutorials*, 16(4):2019–2036, Fourthquarter 2014.

[8] Hongyu Gao, Jun Hu, Tuo Huang, Jingnan Wang, and Yan Chen. Security issues in online social networks. *IEEE Internet Computing*, 15:56–63, 2011.

[9] pleaserobme.com. Raising awareness about over-sharing, 2017.

[10] Dara Hallinan, Michael Friedewald, and Paul McCarthy. Citizens' perceptions of data protection and privacy in europe. *Computer Law & Security Review*, 28(3):263–272, 2012.

[11] Isabell Büschel, Rostane Mehdi, Anne Cammilleri, Yousri Marzouki, and Bernice Elger. Protecting human health and security in digital europe: How to deal with the "privacy paradox"? *Science and Engineering Ethics*, 20(3):639–658, Sep 2014.

[12] Robert Glancy. Will you read this article about terms and conditions? you really should do — robert glancy, 2014.

[13] Emiliano De Cristofaro, Claudio Soriente, Gene Tsudik, and Andrew Williams. Hummingbird: Privacy at the time of twitter. In *2012 IEEE Symposium on Security and Privacy*, pages 285–299. IEEE, May 2012.

[14] Indrajeet Singh, Michael Butkiewicz, Harsha V Madhyastha, Srikanth V Krishnamurthy, and Sateesh Addepalli. Twitsper: Tweeting privately. *IEEE Security Privacy*, 11(3):46–50, May 2013.

[15] JoinDiaspora*, 2017.

[16] Thomas Paul, Antonino Famulari, and Thorsten Strufe. A survey on decentralized online social networks. *Computer Networks*, 75:437 – 452, 2014.

[17] Shihabur Rahman Chowdhury, Arup Raton Roy, Maheen Shaikh, and Khuzaima Daudjee. A taxonomy of decentralized online social networks. *Peer-to-Peer Networking and Applications*, 8(3):367–383, May 2015.

[18] G NaliniPriya and M Asswini. A survey on vulnerable attacks in online social networks. In *International Confernce on Innovation Information in Computing Technologies*, pages 1–6. IEEE, Feb 2015.

[19] Matt Bishop. *Introduction to computer security, 1st ed*. Boston: Addison-Wesley, 1st edition edition, 2008.

[20] Shon Harris. *CISSP all-in-one exam guide*. McGraw-Hill, 2012.

[21] Kim Wuyts. Linddun privacy threat modeling, 2017.

[22] Christos Kalloniatis, Evangelia Kavakli, and Stefanos Gritzalis. Addressing privacy requirements in system design: the pris method. *Requirements Engineering*, 13(3):241–255, Sep 2008.

[23] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, 2010.

[24] Gary Stoneburner. Underlying technical models for information technology security: Recommendations of the national institute of standards and technology, 2001.

[25] Facebook., 2017.

[26] Yoshinori Matsunobu. Semi-synchronous replication at facebook, 2014.

[27] Data Center Knowledge. The facebook data center faq, 2016.

[28] Kate Conger. The facebook data center faq, 2016.

[29] Facebook. Messenger secret conversations. Technical report, Jul 2016.

[30] Facebook. Statement of rights and responsibilities, 2017.

[31] Twitter., 2017.

[32] Aliza Rosen. Tweeting made easier, Nov 2017.

[33] Twitter., 2017.

[34] Peter Schuller. Manhattan, our real-time, multi-tenant distributed database for twitter scale, Apr 2014.

[35] Caitlin Dewey. Ellen degeneres' oscar selfie broke twitter, and world records, Mar 2014.

[36] Twitter. Request to verify an account — twitter help center, 2017.

[37] Twitter. Privacy policy, Jun 2017.

[38] Jaiku archive. Google acquires jaiku, 2007.

[39] Bradley Horowitz. A fall sweep, Oct 2011.

[40] Wikipedians. *Google Services*. PediaPress, 2010.

[41] tumblr., 2017.

[42] Yahoo! Yahoo! to acquire tumblr, 2013.

[43] Todd Hoff. Tumblr architecture - 15 billion page views a month and harder to scale than twitter, Feb 2012.

[44] MariaDB. Tumblr uses mariadb for multi-source replication, 2017.

[45] tumblr. Tumblr privacy policy, 2017.

[46] Plurk, 2017.

[47] Alexa Internet. plurk.com traffic statistics, 2017.

[48] pump.io, 2017.

[49] identi.ca, 2017.

[50] wikipedia, 2017.

[51] DiasporaFoundation, 2017.

[52] Gordon Morehouse, 2017.

[53] twister., 2017.

[54] Miguel Freitas. Twister: the development of a peer-to-peer microblogging platform. *International Journal of Parallel, Emergent and Distributed Systems*, 31(1):20–33, 2016.

[55] Miguel Freitas. twister-a p2p microblogging platform. *arXiv preprint arXiv:1312.7152*, 2013.

[56] trsst., 2017.

[57] GitHub., 2017.

[58] trsst. Trsst: a secure and distributed blog platform for the open web. Technical report, Aug 2013.

[59] GitHub., Mar 2014.

[60] Gab., 2017.

[61] Gab. Happy birthday, gab: Announcing our plans for an ico, Aug 2017.

[62] Gab., 2017.

[63] Andrew Torba. Andrew torba on gab, Sep 2017.

[64] Alan Dreyfus. Gab.ai bans first user, Jan 2017.

[65] Gnu Social, 2017.

[66] Gnu Social, 2017.

[67] World Wide Web Consortium, 2017.

[68] Sonja Buchegger, Doris Schiöberg, Le-Hung Vu, and Anwitaman Datta. Peerson: P2p social networking: Early experiences and insights. In *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, SNS '09, pages 46–52, New York, NY, USA, 2009. ACM.

[69] Leucio Antonio Cutillo, Refik Molva, and Thorsten Strufe. Safebook: Feasibility of transitive cooperation for privacy on a decentralized social network. In *2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks Workshops*, pages 1–6, June 2009.

[70] Leucio Antonio Cutillo, Refik Molva, and Thorsten Strufe. Safebook: A privacy-preserving online social network leveraging on real-life trust. *IEEE Communications Magazine*, 47(12):94–101, Dec 2009.

[71] Leucio Antonio Cutillo, Refik Molva, and Melek Önen. Safebook: A distributed privacy preserving online social network. In *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–3, June 2011.

[72] Daniel R. Sandler and Dan S. Wallach. Birds of a fethr: Open, decentralized micropublishing. In *Proceedings of the 8th International Conference on Peer-to-peer Systems*, IPTPS'09, pages 1–1, Berkeley, CA, USA, 2009. USENIX Association.

[73] Timothy Perfitt and Burkhard Englert. Megaphone: Fault tolerant, scalable, and trustworthy p2p microblogging. In *2010 Fifth International Conference on Internet and Web Applications and Services*, pages 469–477, May 2010.

[74] Kalman Graffi, Christian Gross, Patrick Mukherjee, Aleksandra Kovacevic, and Ralf Steinmetz. Lifesocial.kom: A p2p-based platform for secure online social networks. In *2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P)*, pages 1–2, Aug 2010.

[75] Peter Druschel and Antony Rowstron. Past: a large-scale, persistent peer-to-peer storage utility. In *Proceedings Eighth Workshop on Hot Topics in Operating Systems*, pages 75–80, May 2001.

[76] Kalman Graffi, Christian Gross, Dominik Stingl, Daniel Hartung, Aleksandra Kovacevic, and Ralf Steinmetz. Lifesocial.kom: A secure and p2p-based solution for online social networks. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 554–558, Jan 2011.

[77] Tianyin Xu, Yang Chen, Jin Zhao, and Xiaoming Fu. Cuckoo: Towards decentralized, socio-aware online microblogging services and data measurements. In *Proceedings of the 2Nd ACM International Workshop on Hot Topics in Planet-scale Measurement*, HotPlanet '10, pages 4:1–4:6, New York, NY, USA, 2010. ACM.

[78] Tianyin Xu, Yang Chen, Lei Jiao, Ben Y. Zhao, Pan Hui, and Xiaoming Fu. *Scaling Microblogging Services with Divergent Traffic Demands*, pages 20–40. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[79] Amre Shakimov, Alexander Varshavsky, Landon P. Cox, and Ramón Cáceres. Privacy, cost, and availability tradeoffs in decentralized osns. In *Proceedings of the 2Nd ACM Workshop on Online Social Networks*, WOSN '09, pages 13–18, New York, NY, USA, 2009. ACM.

[80] Amre Shakimov, Harold Lim, Ramón Cáceres, Landon P Cox, Kevin Li, Dongtao Liu, and Alexander Varshavsky. Vis-vis: Privacy-preserving online social networking via virtual individual servers. In *2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)*, pages 1–10, Jan 2011.

[81] DPCS. Garlanet, 2011.

[82] Emiliano De Cristofaro, Claudio Soriente, Gene Tsudik, and Andrew Williams. Tweeting with hummingbird: Privacy in large-scale micro-blogging osns. *IEEE Data Eng. Bull.*, 35(4):93–100, 2012.

[83] Sonia Jahid, Shirin Nilizadeh, Prateek Mittal, Nikita Borisov, and Apu Kapadia. Decent: A decentralized architecture for enforcing privacy in online social networks. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 326–332, March 2012.

[84] Sonia Jahid, Prateek Mittal, and Nikita Borisov. Easier: Encryption-based access control in social networks with efficient revocation. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, pages 411–415, New York, NY, USA, 2011. ACM.

[85] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy (SP '07)*, pages 321–334, May 2007.

[86] Shirin Nilizadeh, Sonia Jahid, Prateek Mittal, Nikita Borisov, and Apu Kapadia. Cachet: A decentralized architecture for privacy preserving social networking with caching. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '12, pages 337–348, New York, NY, USA, 2012. ACM.

[87] Jörg Daubert, Leon Böck, Panayotis Kikirasy, Max Mühlhäuser, and Mathias Fischer. Twitterize: Anonymous micro-blogging. In *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, pages 817–823, Nov 2014.

[88] Jörg Daubert. *Anonymous Publish-Subscribe Overlays*. PhD thesis, Darmstadt, Technische Universität Darmstadt, Darmstadt, Germany, 2016.

Appendix

✓, in the following tables, indicates that the corresponding property is present in the discussed system while ✗implies that the property doesn't exist. N/A means that no information was found about the corresponding property or it wasn't addressed in case of proposed OSNs.

Table 2. Classification of OSNs by the service provided, architecture and storage

System	Service provided	Architecture	Storage
Deployed			
Facebook	Mixed Services	Centralized	Centralized databases
Twitter	Microblogging	Centralized	Centralized databases
Jaiku	Microblogging	Centralized	Centralized databases
Tumblr	Microblogging	Centralized	Centralized databases
Plurk	Microblogging	Centralized	Centralized databases
Pump.io	Microblogging	Federated	Pods
Diaspora	Mixed Services	Federated	Pods
Twitsper	Microblogging	Centralized	Centralized databases
Twister	Microblogging	Decentralized	Locally on user's machine
Trsst	Microblogging	Hybrid	Locally on user's machine
http://gab.ai	Microblogging	Centralized	Centralized databases
GNU Social	Microblogging	Federated	Pods
Not Deployed			
PeerSon	Mixed services	Decentralized	Locally on user's machine
Safebook	Microblogging	Hybrid	Locally on user's machine
FETHR	Microblogging	Decentralized	Locally on user's machine
Megaphone	Microblogging	Decentralized	Locally on user's machine
LifeSocial.Kom	Mixed services	Decentralized	Locally on user's machine
Cuckoo	Microblogging	Hybrid	Centralized databases
Vis-à-Vis	Mixed services	Federated	Pods
Garlanet	Microblogging	Hybrid	Locally on user's machine
HummingBird	Microblogging	Centralized	Centralized databases
DECENT	Mixed services	Decentralized	Locally on user's machine
Cachet	Mixed services	Decentralized	Locally on user's machine
Twitterize	Microblogging	Centralized	Centralized databases

Table 3. Classification of OSNs by encryption mechanism and key management.

System	Encryption Algorithms	Key Management
	Deployed	
Facebook	TLS certificate and AES 256 with CBC	Keys are device specific.
Twitter	TLS certificate	N/A
Jaiku	N/A	N/A
Tumblr	TLS certificate	N/A
Plurk	TLS certificate	N/A
Pump.io	TLS certificate	N/A
Diaspora	Pretty Good Privacy (PGP)	Public key pair generated by users and stored on the pods.
Twitsper	TLS certificate, AES and SHA512	Group key derived from the content of each message.
Twister	ECIS	End-to-end encryption and keys exchanged out of band.
Trsst	AES-256 and ECDH	Usage of session keys encrypted with ECDH .
http://gab.ai	TLS certificate	N/A
GNU Social	TLS certificate	N/A
	Not Deployed	
PeerSon	Public-key cryptography	Not detailed
Safebook	Public-key cryptography	Not detailed
FETHR	Cryptographic measures are used, but not detailed	N/A
Megaphone	RSA	Self-signed key pairs generated by users and public keys exchanged when joining the tree and stored at the level of the poster.
LifeSocial.Kom	Public Keys and Symmetric cryptography	Usage of session keys encrypted with public key of each follower.
Cuckoo	Public-key cryptography	Public keys stored on the server cloud and exchanged out of band.
Vis-à-Vis	Public-key cryptography	Self-signed key pairs generated by the user and exchanged out of band.
Garlanet	RSA and AES	Keys exchanged out of band.
HummingBird	RSA	Keys exchanged out of band and stored on the server.
DECENT	AES for encryption, DSA for signatures, and RSA to encrypt the write policy signature key.	Keys generated by the users and exchanged out of band
Cachet	Attribute-Based Encryption.	Keys exchanged out-of-band.
Twitterize	AES 128bit in CBC mode	Keys exchanged out of band and stored on the database.

Table 4. Classification of OSNs based on the functionalities they provide

System	Edit profile	Profile Visibility	Relations	Follow Interests	Mention	Reshare	Reply	Search	Recommend Interests	Post Sharing	Content Visibility	Instant Messaging	Media Sharing
						Deployed							
Facebook	✓	Public/ private	✓	✓	✓	✓	✓	✓	✓	Public/ private	Public/ private	✓	✓
Twitter	✓	Public	✓	✓	✓	✓	✓	✓	✓	Public/ private	Public/ private	✓	✓
Jaiku	✓	Public	N/A	✓	✓	✗	✓	✗	✗	Public/ private	Public/ private	✓	✓
Tumblr	✓	Public	✓	✓	✓	✓	✓	✓	✓	Public	Public	✓	✓
Plurk	✓	Public/ private	✓	✓	✓	✓	✓	✓	✓	Public/ private	Public/ private	✓	✓
Pump.io	✓	Public	✗	✓	✗	✓	✓	✓	✗	Public/ private	Public/ private	✗	✗
Diaspora	✓	Publice	✗	✓	✓	✓	✓	✓	✗	Public/ private	Public/ private	✓	✓
Twitsper	✓	Public	✓	✓	✓	✗	✓	✓	✓	Private	Private	✓	✓
Twister	N/A	Private	✗	✓	✓	✗	✓	✓	✗	Private	Private	✗	✗
Trsst	✓	Public	✓	✓	✓	✓	✓	✓	✓	Public/ private	Public/ private	✗	✓
http://gab.ai	✓	Public/ private	✓	✓	✓	✓	✓	✓	✓	Public/ private	Public/ private	✓	✓
GNU Social	✓	Public	✓	✓	✗	✓	✓	✓	✗	Public/ private	Public/ private	✓	✓
						Not Deployed							
PeerSon	✓	Public	✓	✗	✗	✗	✓	✓	✗	Private	Private	✓	✓
Safebook	✗	Private	✓	✗	✓	✗	✓	✓	✓	Private	Private	✓	✗
FETHR	✗	Public	✓	✗	✗	✗	✗	✓	✗	Private	Private	✗	✗
Megaphone	✓	Private	✓	✓	✗	✗	✓	✓	✗	Private	Private	✗	✗
LifeSocial.Kom	✓	Private	✗	✓	✗	✗	✗	✓	✗	Private	Private	✓	✓
Cuckoo	✗	Public	✗	✗	✗	✗	✗	✓	✗	Public	Public	✗	✗
Vis-à-Vis	✗	N/A	✓	✓	✗	✗	✗	✓	✗	Private	Private	✗	✗
Garlanet	✓	Private	✓	✗	✗	✓	✗	✗	✗	Private	Private	Und.Dev	Und.Dev
HummingBird	N/A	Public	✓	✓	✗	✗	✗	✗	✗	Private	Private	✗	✓
DECENT	✓	Public	✓	✗	✓	✗	✓	✓	✗	Private	Private	✗	✗
Cachet	N/A	Private	✓	✓	✓	✗	✓	✓	✗	Private	Private	✗	✓
Twitterize	✓	N/A	✓	✓	✗	✗	✗	✗	✗	Private	Private	✗	✗

Table 5. Classification of OSNs by security goals

System	Availability	Confidentiality	Integrity	Identity Creation	Identity Verification
			Deployed		
Facebook	Servers	TLS/Encryption	✗	User's information	Email/phone and password
Twitter	Servers	TLS	✗	User's information	Email and password
Jaiku	Servers	✗	✗	User's information	Username and password
Tumblr	Servers	TLS	✗	User's information	Email and password
Plurk	Servers	TLS	✗	User's information	Username and password
Pump.io	N/A	TLS	✗	User's information	Username and password
Diaspora	Pods	Encryption	✗	User's information	Username and password
Twitsper	Servers	TLS/Encryption	✗	User's information	Username and password
Twister	Replication	Encryption	Digital sign	Unique user ID	Username and password
Trsst	N/A	Encryption	Digital sign	Key pair	Public key
http://gab.ai	Servers	TLS	✗	User's information	Username and password
GNU Social	Replication	TLS	✗	User's information	Username/email and password
			Not Deployed		
PeerSon	Replication	Encryption	N/A	A hash of address	User ID
Safebook	Replication	Encryption	Digital sign	ID generated by the TIS	User ID
FETHR	Replication	N/A	Digital sign	N/A	The canonical url
Megaphone	Replication	Encryption	Digital sign	The hash of username and the public key	Public Key
LifeSocial.Kom	Replication	Encryption	Digital sign	Key pair	Public key
Cuckoo	Replication	Optional	Digital sign	Server assign ID	N/A
Vis-à-Vis	Replication	Encryption	Digital sign	Key pair	Public key
Garlanet	Replication	Encryption	Digital sign	Username and Key pair	Credentials of the users
HummingBird	N/A	Encryption	✗	Handled by the server	✗
DECENT	Replication	Encryption	Digital sign	Key pair	User ID
Cachet	Replication	Encryption	Digital sign	Key pair	Public key
Twitterize	N/A	Encryption	Optional	User's information	Username and password

Table 6. Classification of OSNs by privacy goals

System	Anonymity	Pseudonymity	Unlinkability	Unobservability	Data protection	Data access	User Authorization	System Authorization	Censorship resistance
				Deployed					
Facebook	✗	✗	✗	✗	TLS & Encryption	No control over the data	Limit the access	✓	✗
Twitter	✗	✗	✗	✗	TLS Certificate	No control over the data	Limit the access	✓	✗
Jaiku	✗	✗	✗	✗	✗	No control over the data	Limit the access	✓	✗
Tumblr	✗	✗	✗	✗	TLS Certificate	No control over the data	Limit the access	✓	✗
Plurk	Optional	Optional	✗	✗	TLS Certificate	No control over the data	Limit the access	✓	✗
Pump.io	✗	✗	✗	✗	TLS certificate	Host data on own pods	N/A	N/A	✓
Diaspora	✗	✗	Private list of followers	Only authorized users can see private posts	Encryption	Host data on own pods	Limit the access	The admin of pods	✓
Twitsper	✗	✗	✗	✗	Encryption	No control over the data	Limit the access	✓	✗
Twister	Hidden identity	User ID	Private "following list"	Hard to impersonate	Encryption	Host data locally on own machines	Limit the access	N/A	✓
Trsst	✓	Users IDs	Public list of followers	Public user's ID and posts	Only private messages are encrypted	Open to public	Limit the access	✗	✓
http://gab.ai	✗	✗	✗	✗	TLS Certificate	No control over the data	Limit the access	✓	✗
GNU Social	✗	✗	Public lists of followers	✗	TLS certificate	Host data on own pods	Limit the access	The admin of pods	✓
				Not Deployed					
PeerSon	✗	✗	✗	✗	Encryption	Users manage the data	Limit the access	✗	✓
Safebook	ID is known to the TIS	User ID	Sender and recipient known to direct friends	✓	Encryption	Users manage the data	Limit the access	✗	✓
FETHR	✗	Canonical URL	✗	✗	✗	Users manage the storage of data	✗	✗	✓

Table 6. Classification of OSNs by privacy goals

System	Anonymity	Pseudonymity	Unlinkability	Unobservability	Data protection	Data access	User Authorization	System Authorization	Censorship resistance
Megaphone	✓	Hash of the public keys	✗	✗	Encryption	Authorized followers can access the data	Limit the access	✗	✓
LifeSocial.Kom	✓	Public key-based ID	✗	✗	Encryption	Users manage the data	Limit the access	✗	✓
Cuckoo	✗	✗	✗	✗	Optional Encryption	No control over the data	N/A	✓	✓
Vis-à-Vis	✓	Public key-based ID	✗	✗	Encryption	No control over the data	Limit the access.	The admins of VIS	Admins of VIS can bans users
Garlanet	✓	✓	The friendship is not revealed to anyone	Difficult to correlate data	Encryption	Host the data on peers' machines	✗	✓	✓
HummingBird	✗	✗	✗	✗	Encryption	No control over the data	Limit the access	✓	✗
DECENT	✓	✗	✓	✓	Encryption	Needs of authorization	Limit the access	✗	✓
Cachet	✓	Public key based ID	Partial: The members know the authorized users	✓	Encryption	Each write operation needs authorization	Limit the access	✗	✓
Twitterize	Forward and mix to achieve anonymity	✗	✓	✓	Encryption	No control over the data	✗	N/A	✗