This article has been accepted for publication by IEEE
978-1-4673-7303-6/15 $31.00 © 2015 IEEE
DOI: 10.1109/ICDCSW.2015.30

# Security and Privacy of Distributed Online Social Networks

Sanaz Taheri Boshrooyeh, Alptekin Küpçü, Öznur Özkasap
Department of Computer Engineering, Koç University
İstanbul, Turkey
{staheri14, akupcu, oozkasap}@ku.edu.tr

***Abstract*—Online social networks (OSNs) suffer from various security and privacy problems. The main source of the security problems is the central service provider that observes users' data and relationships. Distributed OSN (DOSN) is an alternative approach where users control their data without having any central service provider. In DOSNs, for the sake of data availability, users replicate or cache data in other users of the OSN. The replica nodes are indeed another kind of service provider in a small scale and with a local view. Therefore, even though decentralization removes the global view of the single provider, it results in having several small ones. By this claim, centralized and distributed OSNs have several common security concerns. Although there exist prior studies discussing and classifying security issues, a fine grained classification of various state-of-the-art solutions is not available. In this paper, we focus on the data privacy, data integrity, and secure social search solutions for centralized and distributed OSNs. Furthermore, we discuss open security problems and concerns, that can be used as future research directions.**

***Index Terms*—Distributed online social networks, data privacy, data integrity, secure social search.**

## I. Introduction

One of the most popular of systems for sharing information is **online social networks (OSNs)** that gained huge public attention in the recent years. While the World Wide Web is a content based system, OSNs are *user based* systems. In OSNs, participating users join the network in order to link with other users and share information. Social relationships have an important role for finding users with similar interests, and for locating content [1]. OSNs like Facebook, Google+, and Twitter have great number of users. Around 80% of users of the Internet visit one of the OSN sites everyday [2].

Well known OSNs have a **centralized architecture** which means that a single service provider manages the whole system [3]. Having a centralized architecture and data aggregation improves usability. For example, having one entity for storing the users' data makes the searching process easy. Also, by having a central service provider, the system is more flexible in terms of updating and extending the network and changing the underlying architecture [3].

Centralized architecture also has its own disadvantages. The data must be uploaded to a cloud based storage, which is under the control of a central authority [2]. This causes serious threats for the user privacy since the service provider can easily obtain the users' private data. It also knows the social graph that represents interconnections among OSN users, as well as user preferences and behaviour within the OSN. These kinds of information are under the risk of any misbehaviour of the service provider [3]. Moreover, such central OSN providers are also interesting and wealthy sources of information for the hackers, and hence are frequently targeted by the attackers.

The security problems posed by the centralized nature of OSNs have motivated the research community to develop alternative OSN architectures. To give OSN users more autonomy in terms of storing and controlling the access rights of their content was the most important motivation [2]. There are many **distributed online social networks (DOSNs)** out of which Diaspora [4] is one of the most popular because of its good privacy preserving design [5]. In DOSNs, information sharing is done without a central server. Most of the DOSNs are based on a peer-to-peer (P2P) overlay network architecture [2] where the peers are users who can act as client and server at the same time. This means that they can request the information of other users and other users can request their data. The users decide where to store and with whom to share their data. Every user is equally privileged participant, and can be the source and destination of the provided information. The main obstacle of decentralization is that users are responsible for their data availability. Users, their friends, or other peers need to be online for better availability. Also, proxy nodes can be used for storing users' data and keeping them available [6].

Beside the differences between centralized and distributed OSNs, in terms of security, they have several similarities. The primary reason for the privacy issues in centralized OSN is the centralized provider that stores and controls user data [7]. To conquer this issue, decentralized architecture has been proposed. Unfortunately, in the DOSNs, because of data availability concern, another type of central service provider appears. Since users cannot guarantee full time data availability by relying on their system's ability and it might not have high uptime, replication and caching are proven techniques to ensure availability [2]. In both cases, users must trust other users for the security issues related to their data. In fact, DOSNs reduce the security risks of one big central provider by distributing them among small ones.

**Contributions.** In this paper, we provide a fine grained classification of common security concerns and corresponding solutions in centralized and distributed OSNs. We present state-of-the-art approaches for enabling security in OSNs considering three important aspects: data privacy alongside access control management, data integrity and secure social

search. In contrast to prior work, we emphasize solutions of data integrity and secure social search, as well as solutions in DOSNs since this kind of networks provide more autonomy to their users. We also address architectural design principles of OSNs in relation to the security aspects.

Our classification of security aspects and solutions in OSNs is summarized in Table I. Data privacy deals with hiding user's data from illegitimate curious third parties while providing access to the legitimate ones. This hiding can be against service provider (in centralized OSNs) or other users of OSN. While data privacy concern is addressed, another problem, namely data integrity, should be considered. When we are sure only trusted friends see our data by the access policies we defined over our data, what will happen if someone forges a message on behalf of us or temper it. The former is data owner integrity issue and the latter is the data content integrity. The solutions regarding these concerns as well as historical integrity and integrity of data relations are discussed in this paper. An important functionality in each OSN is to find and establish new friendships, that can be assumed as social search. Moreover, social search also addresses finding any content in a social network. The security and privacy of social search is important since it reveals some information about the searcher and other entities participating in the search process. For example, if Alice wants to find her old friend Carol, then the relationship of Alice and Carol will be disclosed to service provider, or in the case of DOSN, to the intermediate nodes participating in the search.

There exist prior studies classifying security and privacy issues in OSNs [3], [8], [6], [9]. Some of them review and classify the proposed approaches in the literature [3], [8], [6], and another restates proposed methods [9]. However, data integrity and secure social search are important security issues which are not considered extensively in prior work. Our approach is to provide a fine grained classification for security issues and solutions in OSNs.

The rest of the paper is organized as follows. In Section II, we describe the architectural design principles of OSNs. Section III focuses on user data privacy protection and access control management techniques. Aspects of data integrity problem and the corresponding solutions are described in Section IV. Secure social search aspects and solutions are explained in Section V. Finally, we conclude with a discussion of other concerns and open problems as future research directions.

## II. OSN Architectures

### A. Centralized Online Social Networks

All the users' private and personal data like their relationships, uploaded images, and posts, etc. are observable for the OSN provider. The amount of information available to the service provider and the ability to control them makes it an important source of security problems [6], [5]. The security issues raised by the central service provider are as follows [6]:

- **Data retention**: This issue refers to violation of the information lifetime, which makes the data available longer than intended. Provider takes backups of users' data and when users delete their data, service provider may pretend to delete, but nothing may change from the provider's view.
- **OSN employee browsing private information**: This issue is raised by full accessibility of OSN provider to data that can be misused by the employees of OSN provider.
- **Selling of data**: Advertisers need to know users' interests, habits and, preferences to be able to accurately find the target users who are interested in their products . For this reason advertisers buy users' data from OSNs. In this way, OSNs will be motivated to sell this information to get income.

Two main approaches are used to make the centralized OSN architecture secure:

- Some studies aim to overbear the security problems in existing OSNs, with the idea that it is better to improve the well-designed present OSNs, instead of migrating to a distributed architecture. A prototype Facebook application addressing some security issues of the Facebook platform by *proxy cryptography* has been built [10]. A *virtual private social network* without any collaboration of service provider is made to mitigate the privacy issues of social networking sites [11].
- Other studies propose a framework for a centralized OSN providing additional privacy benefits [12], [13], [14]. Hummingbird [12] tried to improve security and privacy of OSNs which are similar to Twitter. For this purpose, Hummingbird designed a prototype for implementation of Twitter by considering the protection of tweet contents and hashtags from the malicious centralized server. Frientegrity, a framework for social networking applications which is able to detect misbehaviour of malicious service provider, is proposed in [13]. Persona, [14] took the power of OSN providers in the case of determining the accessibility of users data for applications. Indeed, it gave users this autonomy to decide who can see their private data, even for the applications, with fine-grained policies.

### B. Distributed Online Social Networks (DOSNs)

DOSNs can be classified based on system components' organization. There are two main system components: control and storage. *Control* deals with lookup (user and content) and identity management services, and *storage* addresses the data storage and availability. A high level classification extended from [2] is as follows:

- **Structured**: Users participate in a structured overlay, or use a third party structured overlay providing service. In such an organization, queries will be resolved in a limited number of steps. Most of the recent DOSNs use structured organization and distributed hash tables (DHTs) for the lookup service. Prpl [15], Peerson [16], Safebook [17] and Cachet [18] all utilize structured control overlay. Vis-a-vis [19] designed its own structure

| Category | Security aspects/solutions |
|---|---|
| Data privacy | Information substitution |
| | Symmetric key encryption |
| | Public key encryption |
| | Attribute based encryption |
| | Identity based broadcast encryption |
| | Hybrid encryption |
| Data integrity | Integrity of data owner and data content |
| | Historical integrity |
| | Integrity of data relations |
| Secure Social Search | Content privacy |
| | Privacy of searcher |
| | Privacy of searched data owner |
| | Trusted search result |

TABLE I: Classification of security aspects and solutions in OSNs

*distributed location trees*, which provides efficient and scalable sharing.

- **Semi-structured**: Semi-structured DOSN makes use of super peers, which are a subset of all users who are responsible for storing the index and managing other users as proposed in Supernova system [20]. Such a structure may include lookup services and tracking of users up-time to find the best places for replication.
- **Unstructured**: No user in the system store any index, and operations of system are simply done by the use of flooding or gossip-based communication between users [21]. This kind of management has almost zero overhead.
- **Hybrid**: This kind of systems combine the benefits of the two types of organizations mentioned above. As the storage overlay, Cachet [18] uses hybrid structured-unstructured overlay using a DHT-based approach together with gossip-based caching to achieve high performance. In the hybrid organization of structured and semi-structured storage overlay of Prpl [15], users are allowed to store their data in a distributed and unstructured way, and then there is a process per user that federates the distributed storage of each user and act as a super peer. These super peers form a structured overlay of storage. The hybrid control overlay of Cuckoo [22] uses structured lookup for finding rare items, whereas, the unstructured lookup helps with the fast discovery of popular items.
- **Server Federation:** This is another architecture for decentralization of OSN [3]. The main purpose of this architecture is to distribute users' data among several servers which are running on separate storage entity. In this way none of them will have a complete global view of the private data stored in the system.

## III. Data privacy

Data privacy protection is defined as the way users can fully control their data and manage its accessibility (i.e., to determine which part of data being shared with whom). The latter is known as access control management, and can be done by defining different groups with various access levels. A group is a set of users having a common feature (e.g., fans of football). To obtain the aforementioned goals, most of the proposed frameworks studied in this paper use data encryption methods (except [11], [19]). For data privacy protection, the following solutions exist:

### A. Information Substitution

Substitution means replacing real information with fake information. This solution is mostly used for hiding data from the service provider. For example, some predefined settings of OSNs force users to share their information in public (e.g., profile picture and name). In such a case, the user can share some pseudo information with service provider to be shown on his profile page and send the real information only to trusted friends. This information, in the form of XML files, are stored and processed locally on the friends' systems by the use of a browser extension [11]. A variant of this method can be applied for hiding users information from targeted ads. Users' data will be split into smaller parts called *atoms*. Users who trust each other can swap their atoms of the same type, which are associated with a unique index kept in a dictionary. For swapping an atom, its index will be encrypted, and the content of the resulting index will be used for swapping. Dictionary is public and only authorized users will be able to trace swapping results [23].

### B. Symmetric Key Encryption

Symmetric (private) key encryption is a well-known technique for encrypting data. The term of symmetric comes from this fact that the same key is used for both of encryption and decryption [24]. In fact, the key is the shared secret between all parties to access the private data. Since symmetric encryption methods use simpler operations, they have the advantage of running faster in comparison to other schemes. On the other hand, having the symmetric key for both encryption and decryption causes some integrity problems. In order to obtain integrity of data alongside utilizing the speed of this technique, symmetric key encryption is mostly used with the combination of other data integrity methods (see Section IV).

In terms of access control management in the symmetric key encryption systems, we should encrypt our data by the use of a symmetric key and then share it with the users who we want to be able to decrypt our data. For each new group,

a distinct key should be defined. Adding a user to the existing group means sharing the group key with that user. For the revocation, we need to create a new key and re-encrypt the whole data. Of course, if someone already decrypted the data and kept a copy, we cannot revoke that.

### C. Public Key Encryption

In the public-key encryption two different and separate keys are used for encryption and decryption [24]. Based on this reason it is also known as asymmetric encryption. These two keys may seem to be separate, but they are mathematically related. The keys named as **public key** and **private key** (secret key). The former used for encryption and the latter for decryption.

In order to manage users' data accessibility, data should be encrypted under the public keys of all group's members and then sent to them. When a user leaves the group, his public key will be deleted from the list of group members. For joining, the condition is reverse. Systems of Flybynight [10] and Peerson [16] use public key encryption.

### D. Attribute Based Encryption (ABE)

ABE is a kind of public key encryption. In this scheme, some attributes make the secret key related to the ciphertext. For example, assume that there is a set of attributes like 'relative', 'doctor', and 'painter'. One can decide to assign attributes of 'relative' and 'doctor' to one his friends named Alice. To do so, he must create a key containing 'relative' and 'doctor' attributes and give it to her [25], [26], [27]. After that point, Alice will be able to decrypt every message encrypted under the combination of attributes given in her key. The attributes embedded in the encrypted message are implicitly managing the accessibility of that message i.e., defining a group of members who are the exact audiences of that message.

In ABE, each message should be encrypted with an *access structure* defined over a set of attributes. This access structure can be any logical expression over the selected attributes, for instance ('relative' OR 'painter') or ('relative' AND 'doctor'). When the logic operation between attributes is OR, it means that having one of the listed attributes is enough. However, for the AND, the condition is different and having all the attributes is necessary. In ABE, it is enough to do a single encryption operation to construct a new group. Usual revocation methods for ABE use frequent re-keying. To remove the accessibility of a revoked user, the previous data which were accessible by him must be encrypted and stored again. This kind of re-encryptions causes an extra overhead to the access control management of OSN and makes it time-consuming. There exist two kinds of ABE based on the association of access structure with the users' secret keys or with the encrypted messages. In the **ciphertext policy ABE (CP-ABE)**, access structure is determined in the encrypted message and key contains a set of attributes while the condition in the **key policy ABE (KP-ABE)** is reverse. Ciphertext policy has a wide range of usage for supporting data privacy in OSNs such as Cachet [18] and Persona [14] making use of ABE.

### E. Identity Based Broadcast Encryption (IBBE)

In a **Broadcast Encryption (BE)** scheme, there exist a broadcast channel among the list of the recipients [28]. Each user has a private key. The broadcaster selects a group of identities in order to encrypt the messages for them. The broadcaster then transmits the messages to the recipients listed in the channel. The recipients use their private keys for the decryption.

In an **Identity Based Encryption (IBE)** scheme, public keys can be any arbitrary string [29] like email addresses. In such schemes, there is a trusted third party named Private Key Generator (PKG) that produces corresponding private keys.

In **Identity Based Broadcast Encryption (IBBE)** schemes, audiences of a broadcast group can use any identifier string as their public keys [30]. Considering the OSNs, the username or e-mail addresses of the members can be used as their public key for sending encrypted messages. From this point of view, IBBE is more flexible than ABE, since it addresses individual recipients instead of the whole group. Removing a recipient from the list would then have no extra cost. Systems such as [31], [14] also use this encryption approach.

### F. Hybrid Encryption

A hybrid encryption is one which combines the convenience of a public-key encryption with the high speed of a symmetric-key encryption. In such systems, access control management is performed in two phases:

- Symmetric encryption of data by the use of a symmetric key.
- Applying public key encryption under the public keys of all group's members to encrypt that symmetric key.

While many implementations share this hybrid encryption framework [13], [12], there are differences in the choices of the symmetric and asymmetric-key encryption used. In Hummingbird [12], the symmetric key is derived by applying a combination of a **pseudo random function (PRF)** and a hash function on a particular part of message (hashtag). For the key dissemination an **oblivious pseudo random function** protocol must be followed between user and his friends.

Informally, a **PRF** [24] family is a set of polynomial time functions such that no one can distinguish between a function randomly chosen from this set and a function that its output is completely random. A PRF f takes two inputs: a secret s and a variable x, and outputs $f_s(x)$.

An **Oblivious PRF (OPRF)** [32] is a protocol running between two parties, sender and receiver. The goal of the protocol is to compute $f_s(x)$ in a secure way. Receiver is the person who wants to know the value of function f in x and of course he determines x. The sender is the person who knows and determines the secret value of s, so he is able to compute $f_s$ for any input. At the end of the execution of the protocol, receiver will learn $f_s(x)$ from the interactions while sender nothing.

The hybrid structure of the access control lists (ACLs) in Frientegrity [13] is organized in a persistent authenticated dictionary (PAD). Thus, ACLs are PADs, making it possible to access in logarithmic time. Persona [14] uses CP-ABE for

data encryption and PKI to share the keys between friends. Cachet [18] uses a hybrid scheme of symmetric key encryption and CP-ABE: the symmetric key which is chosen randomly for data encryption, must be encrypted with ABE for the set of audiences to make them able to decrypt the data. Another hybrid scheme with combination of public key encryption and CP-ABE is applied to grant friends the ability of adding a comment to a post.

## IV. Data Integrity

A common security issue of centralized and distributed OSNs is data integrity [33], [34], [2]. The data integrity is defined as the protection of data from unauthorized or improper modifications and deletions [24], [35]. For the sake of simplicity, we explain and classify different aspects of data integrity in OSNs by the use of a short and simple scenario. Assume that Bob is organizing a party and wants to invite his friends to the party. Alice receives an invitation letter in a packet from Bob, containing this message: "Come to my party held at my home on Friday". Considering this scenario, the different aspects of data integrity are listed as follows:

- **Integrity of the data owner**: How Alice can be sure that the sender of the message is Bob?
- **Integrity of the data content**: Is the content of the message valid? For example, did Bob really say that there will be a party on Friday at his house?
- **Integrity of data history**: Assume that Bob holds several parties per month, all on Fridays. Alice had been invited to some of them (by receiving invitation letters). Is this invitation letter valid for an upcoming event or has it already expired? Also, was this message delivered to Alice at the proper time or in an even weaker assumption, was this message delivered to Alice at any time?
- **Integrity of the data relations**: Is this message issued for Alice or is it Bob's invitation to someone else but sent to her?

Commonly used methods to protect data integrity are based on digital signatures [36], [17], [16], [21], [18], [15]. A digital signature is a mathematical scheme used by the issuer of a digital message in order to convince the recipient about the integrity of the message. Digital signatures are based on public key cryptography [24]. In most of the cases, the message is signed indirectly. In the other word, first the hash of the message obtained by employing a hash function. After that, the hash of the message is signed by a the digital signature scheme. Hash functions can map inputs with different sizes into a fixed length values [24]. For the sake of saving time and space, signing a hashed message is preferred. Moreover, for security, it is needed that the hash function is collision-resistant; so it is very hard to find different messages with the same hash output.

### A. Integrity of the data owner and the data content

The integrity of the data owner and content can simply be guaranteed by the use of digital signature [36], assuming the public key distribution problem is solved. For the signature verification, it is important to know the valid verification key of each signer. One solution is distributing proper keys out-of-band like physical meeting [16], [13] or transferring the keys via e-mail [19].

### B. Historical integrity

For the data history integrity, one solution is to use *hash chaining* alongside digital signature. In this method, the digital signature must be applied on each entry published by a user, and includes the hash of at least one of his prior posts. This causes a provable partial ordering for his posts [21]. Another solution is to establish a dependency between the timelines of different publishers [21]. In this solution, the publisher adds the hashes of prior events from other participants alongside using the digital signature. In this way, a provable order between their messages will be established.

*Fork-consistent* systems can be used for ensuring historical integrity. In [13], authors proposed object history tree accompanied by a fork-consistency approach. The object history tree data structure addresses historical integrity problem where a malicious service provider or any data storage utility cannot present different clients with divergent views of the system's state. As an example of this scenario, assume the situation where the service provider hides some parts of our friends' updates by providing just a partial view to us. Clients share information about their individual views of the history by embedding it in every operation they perform. As a result, if the clients who have been equivocated by the service provider communicate to each other, they will discover the provider's misbehaviour. In this method, the service provider also digitally signs the root of object history tree in order to prevent the client from later falsely accusing the server of cheating.

### C. Integrity of the data relations

To guarantee the links between two entities in the system, for example a post and corresponding comments, one solution is to embed a proper signing key for signing the comments of that post. The signing key is encrypted in a way that only authorized users can decrypt and use it for posting a comment to that particular post. Corresponding verification key is also located in the content of the post. This verification key can be used to verify whether the comments belongs to the post or not, and also to verify the privileges of the commenter [18]. Each post will contain a different signature key, which enables a different sub-groups of the users to write a comment for different posts.

## V. Secure Social Search

Searching in digital social space is a crucial component of OSNs [37] as the social network users mostly do not prefer to be restricted to the existing friends and they intend to find new friends with common interests. Hence, for supporting social search, disclosing some information about users' profiles is required. The more information that is available, the more accurate the social search results would be. Thus, a trade-off between search capabilities and privacy is raised. While

finding friends is one application of social search, advertising is another kind of searching where an advertiser searches for target users with a related interest to advertise products. Security concerns related to social search can be classified as follows:

- **Content privacy**: Privacy of content addresses information leakage by searching a content. Since the content of searched subject can reveal the interest of the searcher, its privacy must be guaranteed.
- **Privacy of searcher**: Hiding the identity of searcher is an important issue as it is also supported in the existing OSNs like Facebook. For instance, if Alice is searching for one of her old friends, Facebook will list a series of friends close to the criteria Alice is searching for, while none of the listed result persons would be informed about this search. Hiding the identity of searcher from the service provider and other OSN's users who may participate in the searching process (in the DOSNs) is also a concern.
- **Privacy of the searched data owner**: It is important for other users to be able to determine to which extent their data would be available for the system's searches.
- **Trusted search result**: How much trust-able is the result of the search? In the case of finding friends with common interests, one user in the search output may be a better choice among all the results when level of trust and popularity is considered.

Based on the system's objective, different levels of privacy can be applied considering the security concerns. For example, privacy of searcher in an advertising scenario might not be important. Since the advertiser wants to introduce itself to the users, there is no concern about its identity. On the other hand, in the case of finding a friend, the privacy must be guaranteed for most of the security concerns.

### A. Content privacy

**Blind Signatures** can help to provide the privacy of content. Blind signature means signing the document without knowing what the document contains [38]. Hummingbird [12] follows an interesting approach where a signature of a massage's keyword is used as a key to encrypt the message. By considering this idea, anyone who gets the signature on that keyword can also decrypt the message. This method can be used in Twitter for publishing and subscribing. Each subscriber will get the signature on the main keyword (hashtag) of each tweet, by the use of the blind signature, while his interest will not be revealed to the publisher.

### B. Privacy of searcher

A solution to support privacy of searcher is to use **proxy**. In this method, the real identity of users will be replaced by aliases via the proxy server. Since the proxy server knows all the aliases of their users, it can forward messages correctly. Servers cannot see the real names of other servers' users. However, the security of this approach can be under the risk by collusion of proxy servers [3].

**Trusted friends network** is another approach that can be used to support privacy of searcher. In this solution, each user connects directly to trusted friends to forward messages. It will cause a concentric circle of friends around each user, which makes it possible to communicate with the user without revealing identity or even IP address [17]. In the above solution, some relaxation considered that friends of a user are trusted parties. That is, the user is sure that his trusted friends would not cause any security problems by knowing that he is the source of a request.

**Zero Knowledge Proof (ZKP)** alongside using pseudonyms is another solution. By employing the ZKP, one can prove that a given statement is true without revealing any extra information about that statement. In the other word, during a ZKP, the only information that is revealed about the statement is that it is true or not [39]. A user can use a pseudonym while searching in the network, and when (s)he wants to reach a content belonging to another person, (s)he uses ZKP to prove having privileges to access [40].

### C. Privacy of the searched data owner

Regarding the security concern about the privacy of the searched data owner, one solution is to define **resource handler** for data. In this way, every data item has a handler as a reference to that data. For example "Alice's birthday" instead of "26 October 1990". When one is interested in knowing the content of that handler, he must prove himself to the data owner and then get access to the real content [40].

### D. Trusted search result

For finding the best choice among the social search results, one solution works based on the real life assumption that the trust between friends are the means for delivery. It means that if Alice trusts Bob and Bob trusts Sara, then Alice can trust Sara too. The amount of trust assigned to Sara by Alice, based on the search chain from Alice to Sara, is a function of trust levels of every intermediate friend of that chain to the successor friend of that chain [41]. In this way, the target users can be ranked and then chosen.

## VI. Conclusion and Open problems

Popular OSNs have hundreds of millions of active users, and these networks serve new forms of communication, organization, and information sharing. Most of OSNs present the followings functionalities: profile creation, access control management, commenting and social search (finding friend and friendship establishment). These functionalities must be served in a secure manner and they should not affect the privacy of users. Security issues raised in OSNs can be classified into three general categories: Data privacy, data integrity and secure social search. Most of the existing attacks in OSNs threat these categories. In this paper, we reviewed recent solutions of the aforementioned security challenges. However, there exist several security problems which have been discovered but have not been fully solved yet.

- **Implicit information leakage**: Being sure that your data has been shared with authorized users is a different

problem when compared to the problem of recognizing which part of data is sensitive. Certain kind of information can implicitly be derived from published data. For example, a phone number by itself does not contain any important information but it can be shown that the name of phone's owner alongside with the name of his parents can be extracted by having the phone number [42]. It is important to identify what kind of information can be inferred from a published and seemingly simple data. This problem, related to such information which are not explicitly noticeable but implicitly extractable, is called implicit information leakage. To the best of our knowledge, no solution for the implicit information leakage has been proposed so far.

- **Data resharing**: As long as the friends of a user are trustable and do not reshare the data which the user shared with them, no problem will be faced. However, there is no control if they want to reshare the user's data with others. In fact, security and privacy is a collective phenomenon [43]. The main problem is how it would be possible to prevent a user's friends from re-sharing the user's data. This problem is similar to showing a hard-copy of an image to a person while he can see the image under the owner's consent, and he cannot make a copy of it and he has to turn it back to the owner.
- **Privacy preserving advertising**: Another problem is to provide privacy preserving advertising for a service provider storing encrypted data of users in order to get income. It is trivial that the service providers commonly reject a business model that prevents them from users' plaintext data mining for marketing purpose. However, such business model has not been well studied yet. Although there has been some work on privacy preserving advertising systems [44], [45], the development of business models that can support privacy preserving services hosted with third-party providers needs to be investigated further.

There are also other concerns out of the scope of this paper, however there exist other studies that covered these concerns:

- **Protection of data from API** : Online social networks usually employ application systems. In most of them, there is no fine grained policy that can control the access of the application to the personal content. In other words, after the user employs an application, he implicitly gives the application all the accesses to the personal content it wants. A recent study [9] reviews proposed works related to this concern.
- **Network inference**: The network inferences show that it is possible to gain access to the users' information that do not consider explicitly in their OSN's shared data. In some cases, this access possibility proved to obtain even very easily. The concepts of identifying hubs, making link predictions [46] and inferring user attributes have been discussed in [9].
- **Sybil attacks**: Sybil attacks are considered very dangerous for preserving the proper operation of an OSN. In a sybil attack, the reputation system of a network will be subverted by attacker who makes (usually multiple) pseudonymous entities. There are multiple ways in which the attacker can subvert the reputation system. For example, by sending spam, trying to de-anonymize the social network by use of sybil friendship, and simply impersonating other users to accuse them [9].
- **Hiding the social graph**: A social graph represents the users connections among each other, their preferences and behaviour in the social network. Users' relations are source of important information. Some network inferences can be done by user's friends information and interests. Work of [3] has reviewed potential approaches to protect the social graph.
- **OSN anonymization and de-anonymization**: OSN providers publish their data for the research activities in the industry and academia. However, very sensitive information could be revealed explicitly and implicitly from the social graph. There should be an "anonymized" way that let the OSN providers to publish these data sets in a way that minimizes the possible risks for their users. Obtaining the anonymized data, one can reverse the anonymization process and identifies the corresponding nodes to the data set (which is known as de-anonymization). Prior studies of [46], [9] cover this concern.

## REFERENCES


[1] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 29–42, ACM, 2007.

[2] S. R. Chowdhury, A. R. Roy, M. Shaikh, and K. Daudjee, "A taxonomy of decentralized online social networks," *Peer-to-Peer Networking and Applications*, pp. 1–17, 2014.

[3] L. Schwittmann, M. Wander, C. Boelmann, and T. Weis, "Privacy preservation in decentralized online social networks," *IEEE Internet Computing*, p. 1, 2013.

[4] A. Bielenberg, L. Helm, A. Gentilucci, D. Stefanescu, and H. Zhang, "The growth of diaspora-a decentralized online social network in the wild," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pp. 13–18, IEEE, 2012.

[5] A. Verma, D. Kshirsagar, and S. Khan, "Privacy and security: Online social networking," *International Journal of Advanced Computer Research*, vol. 3, no. 8, pp. 310–315, 2013.

[6] M. Beye, A. Jeckmans, Z. Erkin, P. Hartel, R. Lagendijk, and Q. Tang, "Literature overview-privacy in online social networks," Centre for Telematics and Information Technology, University of Twente, 2010.

[7] S. Jahid and N. Borisov, "Enhancing security and privacy in online social networks," Technical Report, Illinois University, 2012.

[8] A. Sattikar and D. R. Kulkarni, "A review of security and privacy issues in social networking," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 6, pp. 2784–2787, 2011.

[9] E. Novak and Q. Li, "A survey of security and privacy in online social networks," *College of William and Mary Computer Science Technical Report*, 2012.

[10] M. M. Lucas and N. Borisov, "Flybynight: mitigating the privacy risks of social networking," in *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, pp. 1–8, ACM, 2008.

[11] M. Conti, A. Hasani, and B. Crispo, "Virtual private social networks," in *Proceedings of the first ACM conference on Data and application security and privacy*, pp. 39–50, ACM, 2011.

[12] E. De Cristofaro, C. Soriente, G. Tsudik, and A. Williams, "Hummingbird: Privacy at the time of twitter," in *Security and Privacy (SP), 2012 IEEE Symposium on*, pp. 285–299, IEEE, 2012.

[13] A. J. Feldman, A. Blankstein, M. J. Freedman, and E. W. Felten, "Social networking with frientegrity: Privacy and integrity with an untrusted provider.," in *USENIX Security Symposium*, pp. 647–662, 2012.

[14] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: an online social network with user-defined privacy," in *ACM SIGCOMM Computer Communication Review*, vol. 39, pp. 135–146, ACM, 2009.

[15] S.-W. Seong, J. Seo, M. Nasielski, D. Sengupta, S. Hangal, S. K. Teh, R. Chu, B. Dodson, and M. S. Lam, "Prpl: a decentralized social networking infrastructure," in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, p. 8, ACM, 2010.

[16] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta, "Peerson: P2p social networking: early experiences and insights," in *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, pp. 46–52, ACM, 2009.

[17] L. A. Cutillo, R. Molva, and T. Strufe, "Safebook: A privacy-preserving online social network leveraging on real-life trust," *Communications Magazine, IEEE*, vol. 47, no. 12, pp. 94–101, 2009.

[18] S. Nilizadeh, S. Jahid, P. Mittal, N. Borisov, and A. Kapadia, "Cachet: a decentralized architecture for privacy preserving social networking with caching," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pp. 337–348, ACM, 2012.

[19] A. Shakimov, H. Lim, R. Cáceres, L. P. Cox, K. Li, D. Liu, and A. Varshavsky, "Vis-a-vis: Privacy-preserving online social networking via virtual individual servers," in *Communication Systems and Networks (COMSNETS), 2011 Third International Conference on*, pp. 1–10, IEEE, 2011.

[20] R. Sharma and A. Datta, "Supernova: Super-peers based architecture for decentralized online social networks," in *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pp. 1–10, IEEE, 2012.

[21] D. Sandler and D. S. Wallach, "Birds of a fethr: open, decentralized micropublishing.," in *IPTPS*, p. 1, 2009.

[22] T. Xu, Y. Chen, J. Zhao, and X. 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*, p. 4, ACM, 2010.

[23] S. Guha, K. Tang, and P. Francis, "Noyb: Privacy in online social networks," in *Proceedings of the first workshop on Online social networks*, pp. 49–54, ACM, 2008.

[24] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC Press, 2014.

[25] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*, pp. 457–473, Springer, 2005.

[26] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pp. 321–334, IEEE, 2007.

[27] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, Acm, 2006.

[28] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology—CRYPTO'93*, pp. 480–491, Springer, 1994.

[29] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in cryptology*, pp. 47–53, Springer, 1985.

[30] C. Delerablée, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in *Advances in Cryptology–ASIACRYPT 2007*, pp. 200–215, Springer, 2007.

[31] F. Raji, A. Miri, M. D. Jazi, and B. Malek, "Online social network with flexible and dynamic privacy policies," in *Computer Science and Software Engineering (CSSE), 2011 CSI International Symposium on*, pp. 135–142, IEEE, 2011.

[32] S. Jarecki and X. Liu, "Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection," in *Theory of Cryptography*, pp. 577–594, Springer, 2009.

[33] T. Paul, A. Famulari, and T. Strufe, "A survey on decentralized online social networks," *Computer Networks*, vol. 75, pp. 437–452, 2014.

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

[35] B. Carminati, E. Ferrari, and M. Viviani, "Security and trust in online social networks," *Synthesis Lectures on Information Security, Privacy, & Trust*, vol. 4, no. 3, pp. 1–120, 2013.

[36] P. Stuedi, I. Mohomed, M. Balakrishnan, Z. M. Mao, V. Ramasubramanian, D. Terry, and T. Wobber, "Contrail: Enabling decentralized social networks on smartphones," in *Middleware 2011*, pp. 41–60, Springer, 2011.

[37] C. Zhang, J. Sun, X. Zhu, and Y. Fang, "Privacy and security for online social networks: challenges and opportunities," *Network, IEEE*, vol. 24, no. 4, pp. 13–18, 2010.

[38] A. Juels, M. Luby, and R. Ostrovsky, "Security of blind digital signatures," in *Advances in Cryptology—CRYPTO'97*, pp. 150–164, Springer, 1997.

[39] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems," *Journal of Cryptology*, vol. 7, no. 1, pp. 1–32, 1994.

[40] M. Backes, M. Maffei, and K. Pecina, "A security api for distributed social networks.," in *NDSS*, vol. 11, pp. 35–51, 2011.

[41] C. Huang, Y. Chen, W. Wang, Y. Cui, H. Wang, and N. Du, "A novel social search model based on trust and popularity," in *Broadband Network and Multimedia Technology (IC-BNMT), 2010 3rd IEEE International Conference on*, pp. 1030–1034, IEEE, 2010.

[42] P. Jain, P. Jain, and P. Kumaraguru, "Call me maybe: Understanding nature and risks of sharing mobile numbers on online social networks," in *Proceedings of the first ACM conference on Online social networks*, pp. 101–106, ACM, 2013.

[43] E. Sarigol, D. Garcia, and F. Schweitzer, "Online privacy as a collective phenomenon," in *Proceedings of the second edition of the ACM conference on Online social networks*, pp. 95–106, ACM, 2014.

[44] S. Guha, B. Cheng, and P. Francis, "Privad: Practical privacy in online advertising.," in *NSDI*, 2011.

[45] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas, "Adnostic: Privacy preserving targeted advertising.," in *NDSS*, 2010.

[46] P. Joshi and C.-C. Kuo, "Security and privacy in online social networks: A survey," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, pp. 1–6, IEEE, 2011.