The Devil is in the Metadata – New Privacy Challenges in Decentralised Online Social Networks

Benjamin Greschbach, Gunnar Kreitz and Sonja Buchegger
KTH Royal Institute of Technology
School of Computer Science and Communication
Stockholm, Sweden
{bgre, gkreitz, buc}@csc.kth.se

***Abstract*—Decentralised Online Social Networks (DOSN) are evolving as a promising approach to mitigate design-inherent privacy flaws of logically centralised services such as Facebook, Google+ or Twitter. A common approach to build a DOSN is to use a peer-to-peer architecture. While the absence of a single point of data aggregation strikes the most powerful attacker from the list of adversaries, the decentralisation also removes some privacy protection afforded by the central party's intermediation of all communication. As content storage, access right management, retrieval and other administrative tasks of the service become the obligation of the users, it is non-trivial to hide the metadata of objects and information flows, even when the content itself is encrypted. Such metadata is, deliberately or as a side effect, hidden by the provider in a centralised system.**

In this work, we aim to identify the dangers arising or made more severe from decentralisation, and show how inferences from metadata might invade users' privacy. Furthermore, we discuss general techniques to mitigate or solve the identified issues.

I. Introduction

As people use Social Network Services (SNS) to organise their social life, privacy issues are an inherent concern in these services. Currently, a user must trust the SNS provider to enforce access rights management, not to misuse the provided content, and to be sufficiently secured against third-party attacks. For today's popular SNS providers, however, "people are not customers, but primarily products" [1]. Their business model is based on targeted advertisements, and they have an infamous history of data leakages and privacy breaches.

In response to these shortcomings, Decentralised Online Social Networks (DOSN) have been proposed. There is a wide range of designs spanning from centralised to decentralised network architectures. In the completely decentralised approaches, the users themselves form a peer-to-peer (P2P) network in order to collaboratively provide the storage and communication infrastructure for the social network service. Access control for published content is enforced by cryptographic means so that users need not rely on policies or the benignity of a central provider. In addition, users keep the physical ownership of their content, which prevents censorship, yields higher resilience with respect to network outages, and facilitates data portability.

When solving the privacy issues of the centralised system by moving to a decentralised design, however, new privacy challenges arise. Simply encrypting the content is not enough to hide all sensitive information from attackers, and although a powerful central provider is not present in these kinds of systems, several other adversary models become relevant.

We remark that although several of the issues raised in this paper have been previously mentioned in the literature, the focus in SNS privacy research has been on content confidentiality and on removing the threat that central SNS providers pose. While this was an important development, we believe that the logical next step is to systematically study the effect of distributing the power of the provider over several entities and examining the possibilities for inferences that persist despite content encryption, including traffic analysis issues in this new context. Failing to protect against even a single one of these threats can lead to serious privacy breaches in an otherwise secure system.

A. Our Contributions

In this paper we highlight the new privacy challenges that arise once a centralised SNS is replaced by a DOSN. Specifically, we systematically discuss possible privacy breaches stemming not from the content itself but from its metadata (like size or structure) or data handling (such as communication flows). Furthermore, we discuss the role of different adversaries in DOSNs. Finally, we summarise approaches to mitigate these problems, including those suggested by proposed DOSN implementations. To the best of our knowledge there is no solution dealing with the whole range of the problems we discuss.

B. Paper Outline

The rest of the paper is organised as follows. After referring to related work in Section II, we sketch the different models of SNS implementations including relevant attackers in Section III. Next, Section IV lists the possible metadata privacy leakages, that is sensitive information which can be

978-1-4673-0907-3/12/$31.00 ©2012 IEEE

inferred even when the content does not leak. Section V discusses countermeasures to approach these new challenges before Section VI concludes the paper.

II. Related Work

Research related to the scope of this paper can be found in mainly three areas: privacy issues in SNS, decentralised online social networks, and metadata privacy in general.

The impact of SNS on their users' privacy has been extensively studied. Gross et al. [2] have identified several threats of SNS usage such as stalking; de-anonymisation of external sensitive sources such as anonymised medical records; identity theft, e. g. by social insurance number reconstruction; user profiling by building a digital dossier and simplified social engineering. Danezis et al. [3] point out that the position of a user in a social network reveals characteristics about the person, such as their status and potential influence reach. Paul et al. [4] underline the consequences of massive central data aggregation in conjunction with an advertising-based business model of major SNS providers. They warn against the risks of direct misuse or unintended leakage of this data that is not appropriately protected and hard to anonymise. Krishnamurthy and Wills [5] show that relevant leaks do occur in practice.

One main approach to address the privacy issues in SNS is decentralisation. Buchegger et al. [6] propose the *PeerSoN* system where (encrypted) content is distributed using a P2P network formed by the users of the SNS. Aiello and Ruffo [7] elaborate on a Distributed Hash Table (DHT) based architectural framework supporting SNS functionality. They propose authentication on the routing level and discuss implementations of SNS requirements such as access control, reputation management and search operations. Cutillo et al. [8] introduce *Safebook*, an architectural approach focusing on communication anonymisation. Content is stored at trusted friend nodes and requests are routed through a mix-network formed by social links to obfuscate information flow. Sharma and Datta [9] describe *SuperNova*, that is based on a hybrid network architecture where highly available "super peers" are used for crucial tasks such as helping new users to join the network. The *Persona* project by Baden et al. [10] proposes the use of an attribute-based encryption scheme for social network operations. Finally, Bodriagov and Buchegger [11] scrutinise the proposals for DOSN-tailored encryption schemes and evaluate their performance for SNS operations.

One instance of a metadata privacy leakage in the context of SNS is mentioned by Anderson et al. [12]. They point out the threat of a friend learning about the existence of content she does not have access to. Chew et al. [13] identify three possible leakages in SNS that are not caused directly by the disclosure of content: entries in the user's activity stream that were automatically generated based on the user's activities (also on third party sites); unwanted linkage of different sets of user data; and identifying inferences by merging social graphs. Traffic analysis, extracting and inferring information from network metadata, (see e. g. Danezis and Clayton [3] for an overview) is one of the attack techniques we consider.

III. Decentralising Social Networks

On an abstract level and following a minimalistic definition (e. g. [14]), we assume an SNS to be merely an integration of user generated content with social relationship information.

The latter is used mainly for access control, data presentation, and friendship announcements. Content comprises all active contributions of a user to the system, static (such as profile attributes) as well as more dynamic ones (such as status updates, text-, picture-, video- or link-posts). It also includes interactions such as comments or simple like-indications in response to posts, enrichments of posts with social links (e. g. *tags* in pictures) as well as asynchronous or synchronous messaging (e. g. private messaging, chats). Timestamped notifications about this data are usually automatically pushed to the user in a "news feed".

In a concrete implementation of such a system, the degree of centralisation of control over user data is an architectural design choice that impacts both possible privacy leaks and types of attackers.

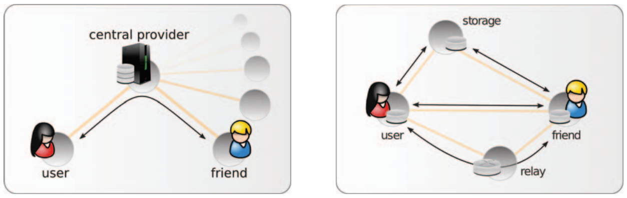
A. Architectures

Considering the proposed DOSN implementations in the literature, one can observe a broad range of topologies rather than a bipartite division between fully centralised and fully decentralised systems. Several hybrid approaches (e. g. *Diaspora*[1]) use dedicated, semi-trusted nodes to address availability, bootstrapping and other issues that are difficult to solve in a flat P2P network. For the rest of this paper, however, we focus on the differences between the two extreme cases of logically centralised designs (*Facebook* or *Google+*) and completely decentralised approaches (*PeerSoN*, *Safebook*, or *Persona*).

In the centralised case, the relevant agents are the SNS provider and the users of the SNS, with all communication between users relayed by the central provider. In the decentralised approach, the provider is replaced by a P2P network formed by the users.

P2P approaches are varied, but here we sketch a simplified example system. A user herself hosts all content she posts. To ensure availability even when she is offline, her content is also replicated by a number of storage nodes. Access control can be implemented either by having the user and storage nodes requiring authorisation to serve data, or by encrypting objects such that only authorised users can decrypt the content. We focus on the latter type of design, where any user can request any encrypted data. This means that storage

[1] http://diasporaproject.org/



(a) All communication is relayed by the central provider.

(b) Besides direct peer communication, several nodes can be involved.

Figure 1. Communication flows in a) centralised and b) decentralised SNS

nodes need not be trusted and need not be informed of ACL rules for the content they replicate.

Another type of service provided by nodes to each other in the P2P scenario is relaying traffic in the overlay. This may include forwarding traffic for two nodes who cannot directly connect due to firewall restrictions, implementing a DHT, or to anonymise communication. We refer to nodes acting in such capacities as relay nodes. We illustrate the different abstract communication flows in Figure 1.

B. *Privacy Advantages of Decentralisation*

The most important privacy advantage of a decentralised system is the absence of a central point of data aggregation. In the case of a centralised system with unencrypted storage, the provider can mine the data without limitations and infer information from both the content and metadata. In addition to deliberate privacy infringements, also by disclosure to third parties, centralised data collections are vulnerable to accidental leaks caused by inadvertent insider behaviour or attacks on the system. In a centralised system that employs user-side encryption to protect the content (e. g. *Pidder*[2]), the provider only observes metadata. When drawing inferences from it, the provider is, however, in the best position possible as it has a complete view of all users of the system at all times.

In a P2P system, data might be replicated by friends (e. g. *Safebook*) or random strangers (e. g. *SuperNova*), but no systematic accumulation of user data occurs.

C. *New Challenges from Decentralisation*

While removing the single point of data aggregation constitutes a general advantage of more decentralised architectures, there are also several drawbacks and new privacy challenges when building on a P2P network. In a centralised system the users' content is entrusted to a single party that only gives access to entitled principals. Deliberately or as a side effect, this intermediation procedure hides metadata information from requesting users. Thus, while the operator of a centralised system can learn significant information from metadata (and content, if not encrypted), such information is hidden from everyone else.

In the decentralised systems we consider, several parties are involved in storing and communicating user content, and authorisation is performed via encryption of the data. This aggravates the problem of metadata privacy leakage because more parties can access such information as illustrated in several examples in Section IV. Unless the decentralised system is carefully designed, it may admit similar privacy invasions from peers in the system or third parties requesting large amounts of data as were possible by the central provider, thus weakening the privacy motivation for selecting a decentralised design.

A new threat that arises from metadata in a decentralised system is that of a more powerful friend adversary (an attacker that exploits its social ties to the user). One feature of the friend adversary contributes eminently to this problem: friends have more background knowledge related to the user – not necessarily acquired only via SNS communication – that enables them to accomplish effective inference attacks even on sparse raw data. If a friend for example knows about a couple of preferred places the user usually visits, coarse IP address based location information suffices to determine the user's exact geographic location with high probability.

Additionally, traffic analysis yields more information in a decentralised system where information is exchanged directly between communicating parties. The intermediation of very high volumes of communication via a few data centres by centralised solutions serves to hide traffic patterns against outside adversaries (but not against the provider).

In a fully decentralised setting, it is also more difficult to enforce a limit on the rate at which data can be requested. This may allow multiple third parties to collect significant amounts of public information from the DOSN. While such information is by definition public, aggregating and indexing a massive amount of it can constitute a privacy invasion.

D. *Adversary Models*

We distinguish between different adversaries in the context of SNS by their functional power resulting from their role and position in the network.

Relay nodes and **storage nodes** can make use of their special role and position in the network. Relay nodes can easily observe all traffic they forward for other nodes, and storage nodes can analyse the data entrusted to them as well as log all requests they receive. **Friends** of a user – or other socially close nodes like friends of a friend – can try to obtain more information than what the user chose to share with them. This can be done by exploiting the way data storage, encryption and communication is implemented in a DOSN. Having additional background knowledge about the user and possibly incentives for targeted attacks can turn a friend into a powerful attacker. Network **sniffers** who observe communication traffic at an arbitrary location in the network constitute another category of possible attackers.

[2] http://pidder.com/

Harvesters are nodes that simply request data from the system to learn from the metadata they receive.

In order to compare the decentralised system architecture with the centralised SNS, we also list the **central SNS provider** as an adversary. If present, it constitutes the most powerful attacker possible because it observes all content and communication from all users of the system. Even if the content is encrypted, the provider still has a complete picture of communication traffic and content metadata.

The adversary types discussed here have access to different data. Here, we consider five categories of data that an adversary may exploit. An adversary may learn **access patterns**, that is information about when content is requested or modified. She may be able to access **ciphertext representations** of content. She may have intimate **background knowledge** about the victim. She may see all or a fraction of the victim's **network traffic**, and be able to relate it to the victim, which we refer to as **micro-scale** network access. Finally, she may have a global but incomplete view of network traffic in the system that we call **macro-scale**.

Table I
ADVERSARY CAPABILITIES

	Relay	**Stor.**	**Friend**	**Sniff.**	**Harv.**	**Cent.**
Access pattern		✓				✓
Ciphertexts		✓	✓		✓	✓
Backgr. know.			✓			
Net, micro	✓					✓
Net, macro				✓		✓

We summarise the capabilities of adversaries in Table I. From this overview, it can be seen that no single class of attacker is as powerful as the central adversary, but unless the system design adequately addresses metadata concerns, the new attackers may be as powerful as the central one. Moreover, it is significantly easier to position oneself as an adversary in a decentralised system than in a centralised one.

Collusions of several agents in the network also need to be considered. This includes a single agent having several of the roles outlined above (e. g. being both a friend and a storage node), and an adversary paying the cost to operate a large number of nodes.

IV. Inferences from Metadata

By metadata privacy leakages we mean disclosures of sensitive personal information that do not stem from the content of published data but from properties of it (such as size or structure) or information generated while managing it (like communication flows).

Possible inferences from metadata can invade a user's privacy in the same way as sensitive personal information obtained from posted content. This includes identifying information (directly or indirectly), general descriptive data (interests, political attitudes, health condition, etc.) as well as more SNS-specific information such as social relationship data (number of friends, nature of relations, etc.), or behavioural data (activity, location, etc.).

While the DOSN approach is a substantial improvement compared to common centralised systems, we want to illustrate which threats to the users' privacy still remain and which new challenges arise. In the following we assume the SNS to be decentralised with all content and communication encrypted. We further assume that it is correctly implemented and perfectly protects the content. Besides that, we only consider a naïve design of the DOSN and individual worst cases in order to give a comprehensive overview of the possible problems. That implies that there are easy fixes for some of the raised issues – this, however, is discussed later in Section V. We have chosen not to study any particular proposed system, as source code for these is not generally available, or only in beta version.

A. *Inferences from Stored Content*

While encrypting the content solves many important privacy concerns, there still remain possibilities of privacy leakages from the stored data. The size, structure, and (implicit) modification time of the ciphertexts may reveal information that the user originally intended to hide. In the following we give examples for each of these properties.

1) Size: The size of an object's ciphertext is an indicator for the content type of the stored object (e. g. like-flag, text, image, video). Additionally, characteristics such as an estimated word count or the length of a video can be inferred. Moreover, the size of larger files, such as video, may be reasonably unique (at least among objects posted during a given time period or from a specific region). Such uniqueness could allow the ISP of a regime sniffing the network to trace which users have re-shared a forbidden video on the DOSN by simply looking for posts of objects with the exact same size as the video at issue. This type of attack requires access to **ciphertexts** or **network** traffic, either **micro-scale** or **macro-scale**.

2) Structure: An adversary may infer not only the size of single objects, but also statistical information about a set of objects of a certain kind, like the number of objects in a list (e. g. unencrypted documents with data references in *Persona*). Linking this knowledge with information about the content type leads to another form of metadata privacy leakage – revealing the number of pictures in an album or the number of comments to a post. This attack requires access to **ciphertexts**.

Assume for example a user sharing pictures of her recent holidays with friends. Being asked for them at work, she decides to grant access to a subset of them to a colleague. Inferring from the data structure that there are more objects in the album than he can decrypt, the colleague learns the exact number of pictures that are hidden from him.

3) Modification History: Once the storage location of a specific object and some general information about its type are identified by an adversary, monitoring the ciphertext for changes reveals possibly sensitive information. The modification history can for example tell something about the frequency of a user's status updates, the intensity of her commenting activity, or other general usage patterns. This attack requires **access pattern** information, or, with polling, access to **ciphertexts**.

Assume a friend observes frequent modifications of an object, identified as the user's encrypted status update representation. While the version displayed to her does not change, the friend learns that she is excluded from at least parts of the user's updates.

B. Inferences from Access Control Mechanisms

One reason for a user to provide social relationship information is the realisation of fine-grained access control mechanisms. Depending on the implementation of these mechanisms, the chosen access right settings might allow conclusions about a user's social relationships to be drawn, as we outline in this section.

1) Encryption Header: If an access control list (ACL) or other cryptographic key material is stored together with the encrypted object – e. g. in a prepended header – the size of this header can allow inferences about the identity or number of individuals who can access the content. This attack requires access to **ciphertexts**.

Exploiting the same feature either for a central object of a user – like her wall representation – or a representative set of content objects belonging to her, can reveal the total number of friends the user has.

2) Key Distribution: Adding a new friend or revoking access rights of an existing friend will – depending on the encryption scheme and implementation – trigger re-keying and/or key distribution mechanisms that can be observed even by users who are not subject to the relationship change itself. This attack requires **micro-scale network** access. When combined with **background knowledge**, significantly more revealing conclusions can be drawn.

Assume a user expels another user from her circle of friends. If a new group key is sent to her remaining friends, an adversary observing this revocation can, together with background information about the user's social relationships, infer the specific person that was removed.

3) Key Reutilisation: If the same key or encryption header is used for several objects, even adversaries who cannot decrypt the content, trivially learn that the same access rights are in place for these objects. This information can be exploited in several ways. Mapping out relations for a large number of objects might allow inferences about the structure of a user's friend circle. A friend, who has access to the objects and observes another user reacting to one of them (e. g. by a comment or a like-flag), immediately learns that this user has access to all the other objects as well. This attack requires access to **ciphertexts**. **Background knowledge** enhances the attack.

C. Inferences from Communication Flows

An adversary can gain additional insights into a user's activities by capturing network traffic that is related to the user. This might be performed by an external network sniffer as well as persons related to the user, e. g. a node that is hosting some of the user's content and observes the access logs.

1) Direct Connections: SNS-related network traffic can already on a very low protocol level (e. g. IP header information) reveal sensitive information. In the case of direct communication with the user's device – a common scenario in P2P architectures – the IP address of this user is trivially obtained and can be tracked over time. This allows correlating with activities of the user on other internet services like file sharing or voice-over-IP (possible even when located behind a NAT, see [15]), determining geographic location information about the user via geo-IP mappings, or inferring general usage patterns, such as the user's online times or working habits (when does the user connect from which device). This attack makes use of **network** access, either **micro-scale** or **macro-scale**. **Background knowledge** allows more precise conclusions to be drawn.

2) Content Requests: The access logs of content that an adversary is hosting or providing to the user disclose the user's requests for specific objects – therefore acting as implicit reading receipts for new content – and might allow general profiling of the user's interests. Moreover, observing a set of users' access patterns has the potential to identify the ownership as well as possible access rights of content objects. Companies may find it profitable to operate a large number of storage nodes in order to monitor requests. For instance, an insurance company may attempt to identify users accessing content posted in groups related to cancer or other diseases. This attack requires **access pattern** information, or **network** access, either **micro-scale** or **macro-scale**.

3) Content Sharing: Storage nodes as well as sniffers that capture traffic to these can easily observe upload activity. This includes the frequency of changes to stored content and might allow similar conclusions as sketched in Section IV-A. Furthermore, timing-based inferences are a possible way to infer access rights if, for example, the distribution of key material to a set of other users is observed shortly after a new content object was uploaded. By monitoring the upload activity of several users, sniffers might moreover learn ownership relations between the stored content and the uploading users. This attack requires **access pattern** information, or **network** access, either **micro-scale** or **macro-scale**.

4) Control Messages: Depending on the protocol implementation, specific user operations such as login, adding friends, search requests, etc. can yield certain patterns of control messages a sniffer can observe and thus infer the kind of operation. The login procedure of a user may comprise polling friends for updates that happened while the user was offline, communicating with storage nodes or similar characteristic sequences of administrative operations. This attack requires **network** access, either **micro-scale** or **macro-scale**.

V. Countermeasures

There exist several approaches to mitigate the described metadata privacy leakages but to the best of our knowledge no comprehensive concept to cope with them all. In the following, we discuss solutions from the DOSN literature as well as from other fields.

A. Stored Content

To hinder inferences based on the size of ciphertexts, padding is one way to obfuscate the exact content length. That might help against fingerprinting objects by size but may still allow inferences about the content type from the order of magnitude. Another strategy could be to split up content objects into blocks of uniform sizes and hide their connection (e. g. [12]). The latter is, however, non-trivial especially against an adversary performing communication flow analysis.

To hide the structure of composite or related storage objects, an encryption scheme that conceals not only the content of the single objects but also indices and links is one solution. To not solely rely on encryption for authorisation, but use it as one of several layers is another approach. If semi-trusted storage nodes perform additional access right validations before delivering encrypted objects, adversaries not involved in storage cannot retrieve the ciphertext or the metadata information. However, this comes with the trade-off that the storage nodes must be given more explicit access-right information about the objects they keep. Additionally, dummy list entries and placeholder values for fixed fields can prevent an adversary from determining if values have been set in a user profile.

Assuming an insider adversary model (e. g. the storage node itself), hiding the modification history of a content object is very difficult. Baden et al. [10] propose to obfuscate the role of a storage object (e. g. a status update document) for that reason. Another way to conceal user-triggered changes can be the introduction of noise in the modification process, but dummy-change operations can be quite costly in terms of performance for an SNS system.

B. Access Control Mechanisms

Most of the presented access control related leakages can be approached with more sophisticated cryptographic schemes. In *Persona* attribute-based encryption (ABE) is used to realise group encryption without encrypting the symmetric content key with the public keys of all recipients. Groups defined by one user can even be reused by other users without them learning the explicit recipient list (and thereby enabling friend-of-a-friend access schemes). The attribute access structure stored with the object, however, might still allow inferences about the audience, e. g. by the attribute names carrying semantic meaning. The *PeerSoN* project suggest to use broadcast encryption schemes that have hidden access structures. Encryption headers in that case do not reveal anything about the audience of the content.

C. Communication Flows

In the literature, several protection mechanisms against communication flow analysis can be found – general ones as well as some explicitly related to SNS. Mix-network like communication anonymisation is a central part of the approach of the *Safebook* project. Information flows are obfuscated by routing them through a mix of socially related nodes, starting by those that are assumed to be most trusted. Caching can also mitigate communication flow leakages by minimising message exchange in general and decorrelating it from specific user actions. Obfuscation by noise – e. g. introducing dummy traffic – comes with the cost of higher network load but might be required in situations where other means are not applicable or not effective. Careful protocol design can help mitigate leakages as well by making control messages indistinguishable from content bearing messages. Another approach is to make the DOSN protocol and communication patterns difficult to distinguish from some existing high-volume P2P protocol, such as *BitTorrent*.

VI. Conclusion

Table II summarises the critical metadata in DOSNs and possible privacy leakages identified in Section IV as well as the approaches to tackle these problems discussed in Section V.

We conclude that while DOSNs have great potential to mitigate inherent privacy flaws of today's centralised SNS, simply encrypting the content is not sufficient. Metadata information like inferences from storage objects, access control mechanisms or traffic has the potential to expose the user to severe privacy threats. Furthermore, new adversaries enter the stage when the SNS has no single provider, as the decentralised network architecture exhibits more diverse points of attack. Some of these attacks are easy to protect against but when implementing a DOSN, these issues have to be considered in a systematic manner in order to offer comprehensive privacy protection.

For future work, we plan to further investigate the special characteristics of the friend-adversary model. The aim is to gain a better insight into which inferences are possible for a socially close attacker in a DOSN setting where only sparse sensitive data but extensive background knowledge is

Table II
Summary of Metadata Leakages

Category	Metadata	Possible Inferences	Countermeasures
Stored Content	Size	content type (image, text), content property (word-count), content fingerprint	padding, uniform block sizes
	Structure	list-sizes (number of comments, number of pictures in an album, ...)	clean encryption headers, two-layer authorisation, placeholder entries
	Modification History	frequencies of status updates, commenting, etc.	noise (dummy change operations)
Access Control Mechanisms	Encryption Header	content audience size or even identities, number of friends	adapted encryption schemes (attribute-based encryption, broadcast encryption)
	Key Distribution	friend status changes	
	Key Reutilisation	same content audience	
Communication Flows	Direct Connections	IP address (usage of other services, geographic location), usage patterns (online times, working habits)	communication anonymisation (mix-networks), caching, noise (dummy traffic), careful protocol design
	Content Requests	reading receipts, interest profiling, information object ownership	
	Content Sharing	upload activities, access rights (by timing-based attacks), information object ownership	
	Control Messages	specific user operations (login, friend management)	

available. Furthermore, we plan to evaluate the efficacy of the discussed countermeasures for concrete DOSN implementations when more mature code becomes available.

VII. Acknowledgements

This research has been funded by the Swedish Foundation for Strategic Research grant SSF FFL09-0086 and the Swedish Research Council grant VR 2009-3793.

References

[1] S. Gürses and al., "SPION D2.1 - State of the Art," in *SBO Security and Privacy for Online Social Networks*, S. Gürses, Ed., 2011.

[2] R. Gross and A. Acquisti, "Information revelation and privacy in online social networks," in *WPES*. ACM, 2005, pp. 71–80.

[3] G. Danezis and R. Clayton, "Introducing traffic analysis," in *Digital Privacy: Theory, Technologies, and Practices*, A. Acquisti, S. Gritzalis, C. Lambrinoudakis, and S. D. C. di Vimercati, Eds. Auerbach, 2007, ch. 5, pp. 95–116.

[4] T. Paul, S. Buchegger, and T. Strufe, "Decentralized social networking services," in *Trustworthy Internet*, L. Salgarelli, G. Bianchi, and N. Blefari-Melazzi, Eds. Springer, 2011, ch. 14, pp. 187–199.

[5] B. Krishnamurthy and C. E. Wills, "On the leakage of personally identifiable information via online social networks," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 112–117, 2010.

[6] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta, *PeerSoN: P2P social networking: early experiences and insights*. ACM Press, 2009.

[7] L. M. Aiello and G. Ruffo, "Secure and flexible framework for decentralized social network services," *IEEE PERCOM*, pp. 594–599, 2010.

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

[9] R. Sharma and A. Datta, "Supernova: Super-peers based architecture for decentralized online social networks," *ArXiv e-prints*, 2011.

[10] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin, "Persona: an online social network with user-defined privacy," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 135–146, 2009.

[11] O. Bodriagov and S. Buchegger, "Encryption for peer-to-peer social networks," in *IEEE SPSN*, 2011.

[12] J. Anderson, C. Diaz, J. Bonneau, and F. Stajano, "Privacy-enabling social networking over untrusted networks," in *WOSN*. ACM, 2009.

[13] M. Chew, D. Balfanz, and B. Laurei, "(Under)mining privacy in social networks," in *IEEE W2SP*, 2008.

[14] A. Lenhart and M. Madden, "Social networking websites and teens: An overview," *Pew Internet project data memo*, vol. 13, 2007.

[15] S. L. Blond, Z. Chao, A. Legout, K. Ross, and W. Dabbous, "I know where you are and what you are sharing: Exploiting P2P communications to invade users' privacy," in *Internet Measurement Conf.*, 2011.