On the Security and Feasibility of Safebook: A Distributed Privacy-Preserving Online Social Network

Leucio Antonio Cutillo, Refik Molva, Thorsten Strufe

On the Security and Feasibility of Safebook: a Distributed Privacy-Preserving Online Social Network.

Leucio Antonio Cutillo*, Refik Molva*, and Thorsten Strufe ‡

* EURECOM, Sophia-Antipolis, France
‡ TU Darmstadt, Darmstadt, Germany
{cutillo,molva}@eurecom.fr, strufe@cs.tu-darmstadt.de

Abstract. Safebook tackles the security and privacy problems of online social networks. It puts a special emphasis on the privacy of users with respect to the application provider and provides defenses against intruders or malicious users. In order to assure privacy in the face of potential violations by the provider, Safebook is designed in a decentralized architecture. It relies on the cooperation among the independent parties that represent the users of the online social network at the same time. Safebook addresses the problem of building secure and privacy-preserving data storage and communication mechanisms in a peer-to-peer system by leveraging trust relationships akin to social networks in real life. This paper resumes the contributions of [7, 9, 8], and extends the first performance and security evaluation of Safebook.

1 Introduction

Having started as a recreational facility, Online Social Networks, like *facebook*, *LinkedIn*, or *Xing* are becoming a predominant player in the global information processing realm both for personal and professional purposes. Catering for a broad range of users of all ages, and a vast difference in social, educational, and national background, they allow even users with limited technical skills to publish personal information and to communicate with one another. The ease of access and increased information dissemination that are inherent features of Online Social Networks (OSN) on the other hand raise new security and privacy concerns for people and companies alike. As the surge of unprecedented network-based security problems that accompanied the global spread of the Internet in the 1990's, the unlimited dissemination of private data through the OSN seems to pave the way for unprecedented data security and privacy exposures. Data and relationships that were strictly confined to the private realm of individuals or organizations are made available to a huge and often unlimited set of parties

This work has been supported by the SOCIALNETS project, grant no 217141, funded by the EC FP7-ICT-2007-8.2 for Pervasive Adaptation

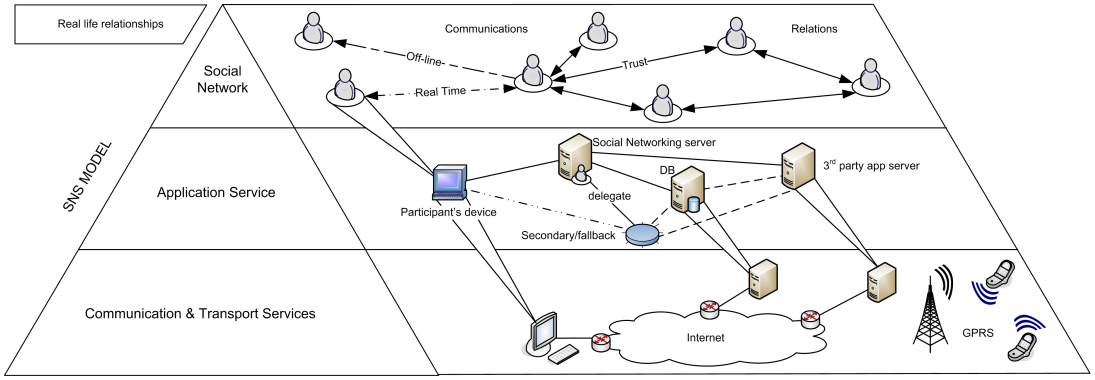
thanks to the facilities of OSN. Access to private data of individuals or organizations becomes much easier for malevolent intruders or simply curious parties either through the lack of restriction by a majority of naive users, the lack of awareness or some breeches in the access control mechanisms of OSN.

Analyzing the OSN with respect to their security properties and the privacy of their users, some obvious threats become apparent. Generally, a wealth of personal data on the participants is stored at the providers, especially in the case of OSN targeting non-professional purposes. This data is either visible to the public, or, if the user is aware of privacy issues and able to use the settings of the respective Social Networking Services (SNS), to a somewhat selected group of other users. As profiles are attributed to presumably known persons from the real world, they are implicitly valued with the same trust as the assumed owner of the profile. Furthermore, any actions and interactions coupled to a profile are again attributed to the assumed owner of this profile, as well. Different studies have shown that the participants clearly represent the weak link for security in OSN and that they are vulnerable to several types of social engineering attacks [12, 4, 14]. This partially is caused by a lack of awareness to the consequences of simple and presumably private actions, like accepting contact requests, tagging pictures, or acts of communication like commenting on profiles or leaving wall posts. However, the usability of privacy controls offered by the SNS, and finally and most importantly, inherent assumptions about other participants and trust in other profiles, which are actually a desired characteristic, certainly add to the problem. However, analyzing the privacy problems in current OSN, it becomes apparent that even if all participants were aware and competent in the use of SNS, and even if a comprehensive set of privacy measures were deployed, the OSN would still be exposed to potential privacy violations by the omniscient service provider: the complete data, directly or indirectly supplied by all participants, is collected and stored permanently at the databases of the providing company, which potentially becomes a big brother capable of exploiting this data in many ways that can violate the privacy of individual users or user groups. The importance of this privacy exposure is underlined by the market capitalization of these providers, of which estimations range from 580m $US, in the case of myspace, to 15bn $US for Facebook Inc. [1]. In consequence, we consider the protection of private data in OSN a pressing topic, which current providers are not likely to address.

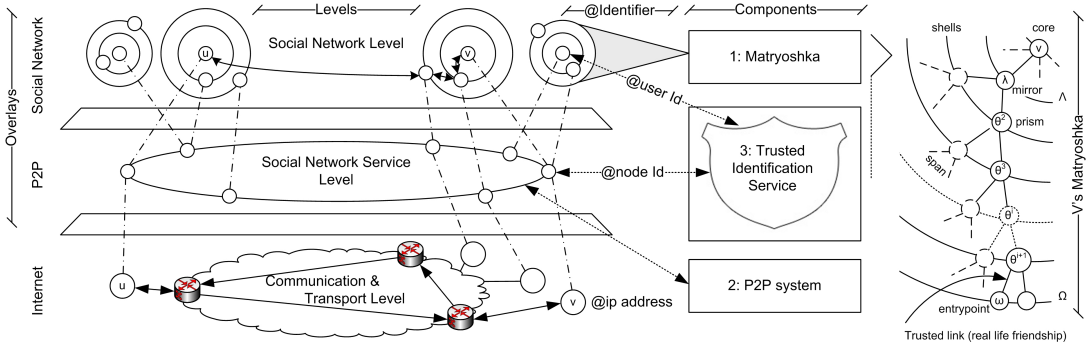
In this paper we suggest a SNS called Safebook that is specifically designed to prevent privacy violations by intruders, malicious users, and OSN providers alike. Safebook is mainly characterized by a decentralized architecture relying on the cooperation among the peers, in order to prevent potential privacy violations due to centralized control.

2 Security in OSN

In order to analyse the security objectives of OSN we first introduce a model to provide a suitable framework for a discussion on their security.



(a) three architectual layers of social networking services



(b) Overlays of Safebook (left), main components (center) and matryoshka structure (right).

Fig. 1. On the top: a model for OSN; on the bottom: an overview of Safebook's architecture.

2.1 SNS Model

Social Network Services can be represented by a layered model (cmp. fig. 1(a)), featuring three levels as follows:

- the ***Social Network*** (SN) level, digitally representing all the users and their relationships;
- the ***Application Services*** (AS) level, hosting the SN application infrastructure;
- the ***Communication & Transport*** (CT) level, providing the classical networking services.

The **SN** level offers a set of functions to the users that are corresponding to interactions in real life, like, e.g., searching for friends, retrieving profile information, displaying information, and giving comments. It typically consists of the actual users and their social interactions provided by advanced services that are based on the SNS infrastructure.
The **AS** level consists of the SNS platform and server that implements the SN functionality as a combination of lower layer mechanisms like, e.g., data storage and retrieval, access control, join and leave management, and is under control of the SNS provider. Various approaches characterized by redundancy and delegation strategies enhance availability to contrast service failures (e.g., redirection of requests to secondary servers in case of high load or server failures). Another part of the AS level are the third party applications that are increasingly made available through the SNS.
The **CT** level finally represents the transport and internetworking protocols and communication infrastructures that provide the basic digital communication facilities.
Based on this model, we define an *internal* attacker as a misbehaving legitimate party, e.g., a malicious user in the SN level, a malicious service provider in the SNS level, or a party that has access to the infrastructure at the CT level, like an eavesdropper with a local-, or a malicious ISP with a global view of the network. An *external* attacker on the other hand is an intruder that tries to violate security at one or more levels (cmp. fig. 1(a)) without the privileges of internal attackers.

2.2 Security Objectives in OSN

Existing threats on OSN raise the three major security requirements of privacy, integrity and availability.
Privacy encompasses a variety of objectives ranging from basic confidentiality, preventing the disclosure of secrets, to the controlled disclosure of sensitive personal data through countering even sophisticated inference techniques that aim at deriving any type of information. Protecting sensitive personal information is especially important in OSN. Privacy threats in OSN include direct information theft by breach of access control schemes or staged attacks, like, e.g., cloning or phishing, that aim at capturing user credentials in order to further disclose

private data. Beyond simple prevention of disclosure, the OSN needs to provide control of the degree at which personal information is disclosed to selected other parties.
The privacy objective often is further detailed into communication unobservability, unlinkability, and untraceability, all of which have to be met, in OSN, too. Unobservability in this case demands that no entity, which is not directly part of the communication can gather any information on request, sender or receiver; unlinkability demands that obtaining two messages, no third party may be able to determine if both messages were sent by the same sender, or to the same receiver; and untraceability finally demands that, given a target user, it should be impossible to list his actions in the system.
Integrity aims at preventing unauthorized modification of information and integrity in OSN focuses on the protection of stored user records against tampering by unauthorized parties, ranging from external intruders to potential internal attackers like maliciously behaving legitimate users. OSN require both the integrity of the data stored in user accounts as well as integrity and authentication as part of the account management. Thus, attacks like profile modification or tampering with data have to be prevented as well as impersonation of legitimate users or cloning of their accounts.
Availability is a global security concern for OSN and aims at assuring the operation of the SNS in the face of malicious or erroneous behavior, preventing users from getting access to the service. The main concern of availability are DoS attacks, but other integrity threats like data pollution and cloning also impair the availability of SNS by affecting the quality of the service perceived by the users.
While privacy has to address broad assumptions regarding adverse parties, including the SNS and application providers as well as external attackers and malicious legitimate users, both integrity and availability primarily address the latter, since the former have an inherent interest that they are met.

3 Decentalized OSN

The architecture of Safebook consists of two overlays, as shown in fig.1(b). Each Safebook node is thus part of the Internet, the peer-to-peer overlay and the social network overlay. The components of Safebook (cmp. fig.1(b)) are:

1. several ***matryoshkas***
2. a ***peer to peer substrate***
3. a ***trusted identification service*** (TIS)

Matryoshkas are particular structures providing end-to-end confidentiality and distributed data storage with privacy. They leverage on existing trust of OSN members in real life. The Peer-to-peer substrate provides a decentralized global data access. The trusted identification service guarantees authentication and provides unique addresses to each member of Safebook. It can be provided off-line and may be implemented in a distributed fashion.

Matryoshkas The Matryoshka of a user (cmp. 1(b) on the right) is a structure composed by various nodes surrounding the user's node in concentric shells. The user's node is thus the ***core*** of his matryoshka and can also be part of some other users' matryoshkas. Every core is associated to a unambiguous ***user identifier*** computed and certified by the TIS. User identifiers are used to route requests through the matryoshkas. The inner shell of a matryoshka consists of nodes belonging to the trusted contacts of the user. The second shell consists of nodes that are trusted contacts of nodes in the inner shell and so on. It is important to note that nodes on the same shell do not necessarily share trust relationships between themselves, except for the inner shell, which all share their relation to the core node.

The nodes on the inner shell cache the data for the core and are thus called ***mirrors***, they serve requests if the core is offline. A data request message reaches a node in the inner shell from a node in the outer shell through a path that provides hop-by-hop trust. The reply follows the same path in the reverse direction. As they act as a gateway for every request to the matryoshka's core, the nodes in the outermost shell are called ***entrypoints***. All the nodes between the mirrors and the entrypoints are called ***prisms*** and extend the hop-by-hop trusted paths. Based on this, the matryoshkas assure cooperation enforcement in our OSN. We point out that the trust relationship between nodes is not used in a transitive fashion, as none of the nodes on a path, other than the direct neighbors, needs to be trusted by any user.

Peer-to-peer substrate The peer-to-peer substrate consists of all the nodes and provides data lookup services. Currently, a DHT derived from KAD[13] is used as the P2P substrate. Nodes are arranged according to their ***node identifiers*** and lookup keys correspond both to members' user identifiers and to the hash of their attributes, like full names or the likes. All nodes that belong to the outer shell of a user's matryoshka register themselves as entrypoints for this matryoshka with the nodes that are responsible for the respective lookup keys. The identity of a peer is revealed only to his trusted contacts since they are the only ones that can link his IP address to his user identifier.

Trusted identification service The trusted identification service (TIS) guarantees resistance against sybil and impersonation attacks by providing each node with a unique node- and user- identifier, and the related certificates. The existence of the TIS does not contrast our goal of privacy preservation through decentralization since the TIS is not involved in any data management activity and it is used only to prevent impersonation and a free selection of a node identifier and hence their position in the DHT. Moreover the TIS can be implemented in a decentralized fashion and does not have to be constantly online.

3.1 Operations

The most important operations of our OSN are the matryoshka creation, the profile publication and the data retrieval.

Matryoshka creation In order to join Safebook a member $\mathcal{V}$ has to be invited by another member $\mathcal{A}$. After this phase, having obtained the necessary credentials from the TIS, $\mathcal{V}$ can start building his matryoshka. $\mathcal{V}$'s final goal is to register in the DHT his user id and a particular set of lookup keys associated to his identity, as e.g. a hash of his full name[1]. At the beginning $\mathcal{V}$ has only $\mathcal{A}$ in his contact list, so he sends $\mathcal{A}$ a signed registration request containing the lookup key(s) he wants to register, his certificate associated to his user id signed by the TIS, and a time-to-live (ttl) counter. This first message presents the user id of the sender instead of his node identifier. This prevents the node in the DHT responsible for $\mathcal{V}$'s lookup key from linking that key with $\mathcal{V}$'s node identifier.

Once $\mathcal{A}$ receives the registration message it decreases the ttl counter, chooses one (or several) of his trusted contacts, called $\mathcal{B}$, as a next step and sends $\mathcal{B}$ the request message signed with his node identifier. This will prevent the registering node in the DHT from retrieving the social relationships between the OSN members constituing $\mathcal{V}$'s matryoshka. It is important to note that no assumption is held about social relationship between $\mathcal{V}$ and $\mathcal{B}$. This process runs until the ttl counter expires, when $\mathcal{V}$'s lookup key is registered in the DHT. The node responsible for that key, hereafter called ***dock***, maintains a reference table associating the key with the ip addresses of the entrypoints of $\mathcal{V}$'s matryoshka.

The number of contacts each node chooses to forward the registration request is determined by the ***spanning factor***. It defines the branching of the tree through the matryoshka whose root are the mirrors and whose leaves are the nodes in the outer shell, starting from the core's direct connections. The higher the spanning factor, the higher is the number of nodes composing the tree, and the higher is thus the probability to have a *valid path* through the tree, i.e. a path where all the nodes are online. The spanning factor and the number of inner shell nodes each core should have is fundamental to guarantee data availability and will be investigated in section 4.

Profile publication A user's data can be public, protected or private and its publication takes place at the contacts' nodes being in the inner shell of the user's matryoshka. All the published data is signed by the owner and encrypted using a simple group-based encryption scheme.

Each node can manage the profile information, the trusted contact relations and the messages. The profile information consists of the data a member wants to publish in the OSN and is organized in atomic attributes. The trusted contact relations represent the *friend list* of the user and associate each contact with a particular trust level. Real time communication messages can be exchanged by each member of the OSN, in this case the communication doesn't stop at the first matryoshka shell but reaches the core.

Data retrieval The requests are routed according to the P2P protocol until they reach a dock. Unlike the common KAD approach, the requests are routed in a recursive way to hide the real requester's node identifier. The dock sends back

[1] $\mathcal{V}$ can of course choose to register different lookup keys, in addition to his user id, to increase his visibility.

the list of all the entrypoints of the target user's matryoshka. The requesting node then sends its request (or delegates a trusted contact to do that) to a subset of the entrypoints of the target matryoshka. The requests are forwarded through the matryoshka to the mirrors, who serve it and send a response along the inverse path. See figure 3(a) for more details.

4 Feasibility

In this section we will analyze the feasibility of our approach with respect to data availability and delays.

We will focus on:

- the minimum number of contacts a node needs to have in order to guarantee the availability of his data;
- the minimum number of hops in the matryoshkas to provide anonymity;
- the expected delay for data retrieval.

Data availability One can see each mirror as a root of a tree whose leaves lie in the outer shell. Let *nop* be the probability of each node being online, *span* the spanning factor of the tree passing through a user $\mathcal{V}$'s matryoshka and *shell* its shells number, i.e. the number of hops between $\mathcal{V}$ and whichever node in the outer shell. Let Λ be the mirror set and $\|\Lambda\|$ its cardinality. Thanks to a simple geometric law (1) it is possible to compute the probability ov_{shell} that at least one inner shell node can be reached, i.e. the probability that $\mathcal{V}$'s data is accessible.

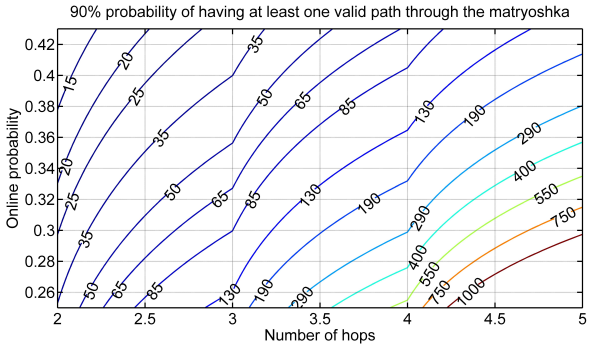
$$\begin{aligned} ov_0 &= nop \\ ov_j &= nop\left(1-\left(1-ov_{j-1}\right)^{span}\right), j \in [1 \dots shell-1] \\ ov_{shell} &= \left(1-\left(1-ov_{shell-1}\right)^{\|\Lambda\|}\right) \end{aligned} \tag{1}$$

Let the probability to have at least one valid path through a user's matryoshka be as high as 90% as a requirement. We refer to a valid path as a path where each node is on-line. Assuming that $span = 1$, this goal is achieved with different values of *shell*, *nop*, and number of contacts in the inner shell, as shown in figure 2(a).

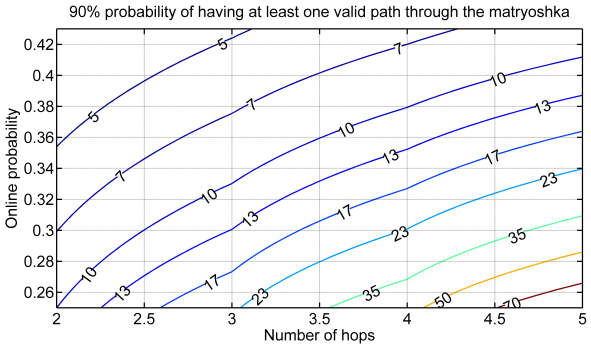
According to a recent work on Skype[2][10] we can assume *nop* to be at least as high as 0.3. We rely on this data since Skype, as Safebook, enhances users' interactions by providing messaging services such as chat.

As one can see in figure 2(a), the number of mirrors λ that is needed with $shell = 3$ and $nop = 0.3$ is 85. With $shell = 4$ the number of mirrors increases to 290. By selecting a spanning factor of $span = 2$, the same availability is achieved with 13 to 23 mirrors, respectively with $shell = 3$ and $shell = 4$ (see figure 2(b)). This amount of contacts is much more likely to be reached. From previous studies we have access to the graph of Xing[3] and could show that the average number η of a member's contacts in that application is 24.

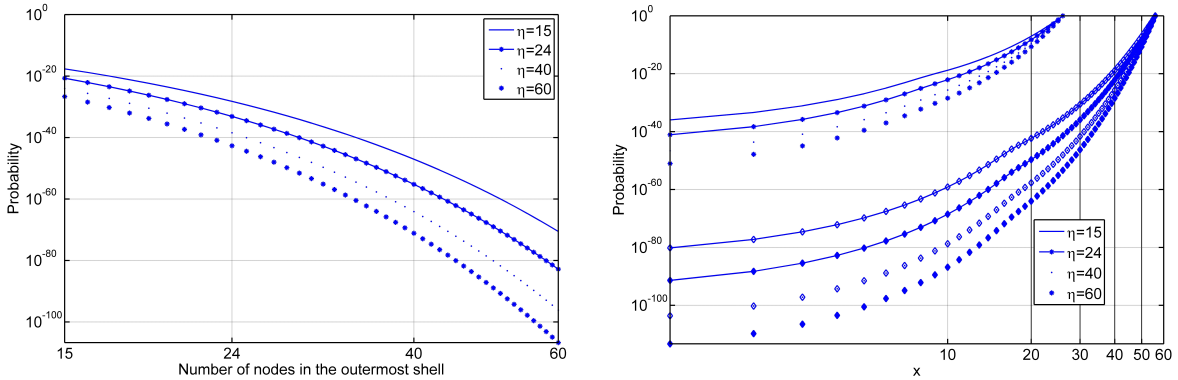
[2] http://www.skype.com
[3] http://xing.com



(a) Access data of a user - span=1.



(b) Access data of a user - span=2.



(c) Disclosure probability over the number of entrypoints with span=1 (left), and over x (right, logscale, the attacker knowing 50%+x of the entrypoints' contact lists), span=2, for $\|\Omega\| = 50$ (top, with points) and $\|\Omega\| = 110$ (bottom, with diamonds).

Fig. 2. On the top: required number of mirrors to guarantee a reachability of 90% for different number of shells (x axis) and online probabilities (y axis); on the bottom: disclosure probability for the identities of the nodes on the first hidden shell.

Data lookup The overall data lookup time T_{dr} can be seen as the sum of the DHT lookup time T_{DHT} and the round trip time in the matryoshka T_M: the first one depends above all on the DHT, while the second one depends above all on the availability of nodes constituing the matryoshka itself.

The choice of the P2P substrate plays an essential role in our OSN performances since it determines T_{DHT}. We use a DHT similar to Kademlia [13] called S2S. Unlike Kademlia, in S2S lookups are performed in a recursive way and message confidentiality is assured with hop-by-hop signature and encription operations.

The round trip time in the matryoshka T_M can be seen as twice the time required to reach a mirror from an entrypoint. As we have shown in the previous sections, a number of hops between three and four reasonably guarantees to each member both anonimity and data availability. This number of hops is comparable with that one encountered, on average, for a successful lookup in KAD[4].

Starting from a CDF representing a one-hop RTT distribution computed from real measurements [16], assuming to find at least one path in S2S where all the nodes are online, we derived the CDF distribution of the total delay for a profile data retrieval in Safebook, taking into account 4 hops for a successful lookup in S2S, 4 hops to cross the matryoshka and one additional hop in case the real data requester delegates the data request to a trusted contact (cmp fig.3(b)). Results show the 90% of profile data lookup succeed in about 10 seconds if no off-line node is met along the path.

Assuming the entrypoint list of the target user's matryoshka is cached, only 5 hops are required and the 90% of future profile data lookup will succeed in about 6 seconds.

Overall data lookup time T_{dr} is thus likely to be on the order of 6-10 seconds, without taking into account that the social proximity can correspond to the geographical one.

5 Security and privacy

The following section discusses Safebook's properties with respect to the privacy, integrity and availability goals we introduced in the first part of this work.

5.1 Separation of Identifiers

In order to protect the privacy, users need to have control over the disclosure of their data to only trusted users. However, to provide the P2P functionality, node ID and IP address of all nodes need to be public and can not be hidden from other participants. Safebook in consequence seperates these two identifiers. While the node ID is used as an address in the P2P overlay, and the node ID public keys are used for hop-by-hop message encryption, the user ID is used to address the

[4] According to recent studies [16] conducted on KAD as implemented in aMule, 90% of the lookups succeed in less than four hops.

users in the social network layer and the user ID public keys in consequence are used for end-to-end encryption of messages between communicating users.Only trusted contacts of a node are able to link these two identifiers, as they serve as mirrors and in consequence know both.
Furthermore, due to the recursive nature of the Safebook protocols, no node inside or outside the matryoshka can trace the trusted connections between two users that span the matryoshka.

5.2 Trusted Identification

A wide range of attacks on P2P systems and online social networks are possible due to the lack of trusted identification of participans. Safebook harnesses the concept of an identification service to this end: The TIS and the certification policy play an essential role in preventing malicious users from manipulating identifiers and performing attacks such as profile cloning, profile porting, identity theft, DoS by aimed placing of nodes in the DHT, sybil and man-in-the-middle.

5.3 Separation of Identification and Communication

The only party in Safebook that is able to link the user ID and node ID of users other than their own trusted acquaintances on the SN level is the TIS. Considering correlated compromise of a TIS by an attacker, which due to misconfiguration logs all registration requests, this ability could potentially be used to break the privacy of Safebook users, by disclosing their participation in Safebook or retrieving their set of trusted contacts. However, the TIS does not possess the keypairs of the user- and node ID and in consequence retrieving profile information does not lead to any information being disclosed, as it is encrypted for trusted users. It is unable to compromise integrity by tampering with messages for the same reason.
Another possibility for disclosure is monitoring the communication relations of nodes. However, the TIS does not participate in any of the communication protocols other than the identity creation and in consequence can not obtain any information as an insider. Analysing the OSN model in section 2, another possibility for monitoring becomes apparent: a collusion of the TIS with the service provider on the CT level would circumvent the concept of separation. However, this attack is only successful if the ISP controls the access to all users of Safebook, as only the privacy of users using the directly monitored Internet connections can be disclosed. Entirely protecting the privacy on the CT level is only possible when leveraging much more complex concepts of anonymization, which for the sake of efficiency is refrained of. Safebook indeed does not provide anonymous communications on the network level.

5.4 Communication Indirection and Cooperation Incentives

Matryoshkas provide the basic OSN services like profile data storage and communication obfuscation, as described in section 3. The caching of profile information

is necessary for reasons of availability, and selecting trusted users for this services leads to an inherent cooperation enforcement. It causes the need to obfuscate who is serving a profile information request, in order to protect the trust relation between the source and the caching node, though. For this reason, several shells of indirection obfuscate the connections and communication between users. Friendship relations between nodes on adjacent shells build hop-by-hop trusted paths for anonymization. The trust in each hop additionally provides cooperation enforcement for the service of forwarding messages, as dropping messages potentially harms the service of a trusted acquaintance.

5.5 Matryoshka Analysis

Considering that the matryoshkas are created based on trusted links, and considering further that humans tend to accept friendship requests and disclose their contact lists more freely than they should [4], it seems feasible to obtain the wealth of relationship information that is innate to the matryoshkas.
Let θ_i^j be the i-th node in the j-th shell of a user $\mathcal{V}$'s matryoshka $\Theta_\mathcal{V}$, with M representing the outermost shell. Let $\{NId_{\theta_i^j}\}$ be its node identifier and $\{UId_{\theta_i^j}\}$ its user identifier. Finally, let $\Omega_\mathcal{V}$ be the entrypoint set of $\mathcal{V}$'s matryoshka. Assuming $\mathcal{U}$ is a malicious user that aims at guessing the relationship information from a selected matryoshka, and $\mathcal{A}$ is a direct contact of this matryoshkas's core $\mathcal{V}$, Safebook is required to hide the information about the relationship between $\mathcal{A}$ and $\mathcal{V}$. The multitude of layered shells prevents $\mathcal{U}$ from directly disclosing another user $\mathcal{A}$'s identity and, as a consequence, $\mathcal{A}$'s friendship with $\mathcal{V}$, as described above. However, $\mathcal{U}$ could try to guess the identity of the nodes and access their contact lists by befriending them from the outer layer through to the core consecutively. Assuming $\mathcal{U}$ retrieves $\Theta_\mathcal{V}$'s entrypoint list $\{NId_{\theta_i^M}\}, \theta_i^M \in \Omega_\mathcal{V}$, $M = Maxshell$, further assuming $\mathcal{U}$ was by chance able to derive all user IDs of the containing nodes, and finally assuming $\mathcal{U}$ gathers that $\Theta_\mathcal{V}$ has $Span = 1$. In this unlikely case the probability for $\mathcal{U}$ to disclose the identity of the prisms $\{UId_{\theta_k^{M-1}}\}$ based on $\{NId_{\theta_i^M}\}$ and by accessing all contact lists of the θ_i^M would be: $(\frac{1}{\eta})^{\|\Omega\|}$ where η represents the average number of contacts of every user. Figure 2(c) (left) plots this probability over the number of entrypoints, showing that it is negligible even for very small values of η.
The task of guessing for an attacker is a little easier when $Span = 2$. In this case, two nodes on a shell share the same predecessor, and $\mathcal{U}$ could derive the cut set of contact lists it obtained, thus generating a good estimate for some of the nodes on the next shell. $\mathcal{U}$ hence needs access to valid contact lists of at least one half of the entrypoints, only, while every additional friend list will improve the chance for correct guesses. If $\mathcal{U}$ can obtain contact lists from $50\% + x$ of the entrypoints, it can compute the intersection between ω_i's and ω_j's friendlists, and, in the worst case that both nodes only share one common contact, derive the identity of one node θ_k^{M-1} with certainty. The probability for the full disclosure of the identities of all predecessors is:

$$\left[1-\left(1-\frac{2^{\frac{\|\Omega\|}{2}}}{\binom{\|\Omega\|}{\frac{\|\Omega\|}{2}}}\right)^{\binom{\frac{\|\Omega\|}{2}}{\frac{\|\Omega\|}{2}+n}}\right]\left(\frac{1}{\eta}\right)^{\left(\frac{\|\Omega\|}{2}-x\right)}$$

Assuming that, in the case of $x = \frac{\|\Omega\|}{2}$, $\mathcal{U}$ has access to all the entrypoint's contact lists, and further assuming the worst case in which the intersections always contain a single node only, $\mathcal{U}$ would thus derive $\left\{UId_{\theta_k^{M-1}}\right\}$ with certainty. Figure 2(c) (right) shows the probability of guessing $\{UId_{\theta^{M-1}}\}$ as a function over x in case $\|\Omega\| = 50$ (top) and $\|\Omega\| = 110$ (bottom), always considering the worst case of atomic intersections. While for low x the probability again is quite low, it unsurprisingly increases to a possibility of 1 with x growing to $\frac{||\Omega||}{2}$.

However, in Safebook neither the number of shells nor the span value are fixed, as the nodes in the registration paths decrease $TtlMatr$ by 1 or more. They additionally can select, according to their characteristics, a number of next hops that slightly differs from $Span$. As finally the barrier for the attacker, which has to obtain both the user ID of the entrypoints and their contact lists are quite high, and as the probability is only valid for the worst case of seperate, atomic intersections between all pairs of contact lists of these users, we consider this vulnerability as negligible.

6 Related work

The fact that OSN pose as a serious threat to the security of their users has been shown in multiple studies[12, 4, 14]. It has sparked a plethora or ideas on how to solve this problem at the same time.

NOYB[11] is an approach that tries to mitigate the existing problems by cryptographic means. Applying substitutions according to secret dictionaries, it renders the managed public profiles, which still may be stored in centralized OSN, useless to anybody lacking access to the dictionaries. While protecting some of the content of the profiles, it does not protect the relation between users, be it an accpted friendship or message exchange.

Yeung et al.[17] propose to use a Friend-of-a-Friend as an OSN: storing contact list information in addition to conventional content at a common webserver, which is maintained by the respective user itself, they provide a framework to create relations between the managed sites, thus indirectly offering OSN functionality. To somehow protect the content partially, they propose some access control based on an existing language for the definition of AC policies. Unlike Safebook, the system does not protect the identity of its users.

Persona [2] is an approach to combine attribute-based crypto with traditional public-key cryptography, to offer a more flexible and fine-grained access control. Persona users are identified by public keys and they store their encrypted data with their own storage service. In order to create an OSN link, they exchange their public key and storage service location out of band. While better protecting the identity of users, the complete privacy that Safebook offers is still not given.

The related work closest to Safebook is probably PeerSon [5]. Buchegger et al. propose to use an existing, external system, OpenDHT, to store the profile information, and encryption to prevent unauthorized access. While PeerSon represents a fully distributed OSN with a much lighter architecture, the privacy protection of Safebook is by far more comprehensive.
An entirely different family of systems is based on a different history, but similar to Safebook: Darknets and related P2P systems[15, 6, 3] aim at anonymizing the communication between their users completely. They follow concepts similar to Safebook: they establish connections between trusted users only and apply hop-by-hop anonymization. However, they typically suffer from delays that are far beyond acceptable for an OSN, are unable to guarantee the availability of less popular content, and do not provide means for any kind of social networking services.

7 Conclusions and Future Work

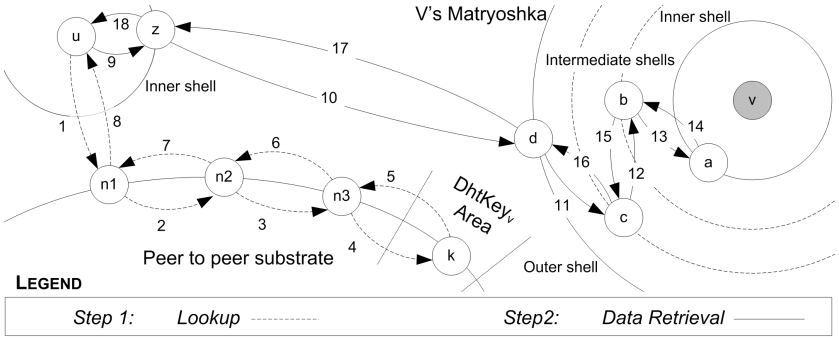
This paper studies the privacy problems that users of current Online Social Networks (OSN) are facing. It defines a layered model to illustrate different parts of a typical OSN infrastructure, the different roles of the participating stakeholders, and possible points of intereference. The model then is used to define security and privacy objectives that Social Networking Services (SNS) are expected to meet.
Since current OSN do not comply to these objectives, which, due to the deviating interest of their stakeholders is not likely to change any time soon, the paper subsequently introduces Safebook, a new approach for privacy preserving online social networking. Safebook is based on the two main ideas of decentralization and leveraging trust from real world relationships. It integrates the three core concepts of the matryoshka, a group of nodes per user, which collaboratively stores the profile information and serves for communication anonymization, a peer-to-peer substrate for the location of the users and their published content, and trusted identification service to guarantee the authenticity of credentials.
In order to evaluate the privacy protection provided by Safebook, it's security properties are subsequently analyzed and discussed in detail. The evaluation shows that Safebook is able to preserve the privacy of its users, even in terms of communication unobservability, untraceability and unlinkability. Additionally it is demonstrated, that Safebook provides integrity and availability.

The decentralized design of Safebook, and the introduction of additional indirection for reasons of communication anonymization through the matryoshka are challenging when considering performance requirements. After performing a preliminary feasibility study, we currently analyze the performance of Safebook in appropriate detail, while being in the process of conducting a comprehensive simulation study to both validate the performance and parametrize the protocols at the same time. In parallel, we already have built a prototype of Safebook, which currently is in the stage of early testing and shall be available for download soon. For the purpose of enhancing the authentication of Safebook, we are planning to

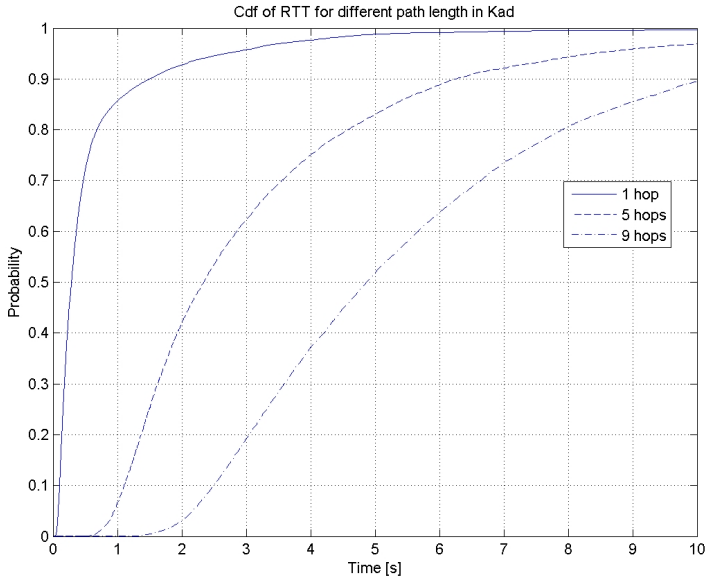
put a stronger focus on the possibilities to better leverage the knowledge from existing trust relationships, quite possibly by applying secret matching schemes and secret handshakes, and to study the interdependency when introducing a reputation scheme into Safebook. The last point promises to be especially interesting, as the assumption of anchoring the participants and their connections in the real world and the relationship between the users significantly changes the setting for decentralized reputation systems.

References

1. Modelling The Real Market Value Of Social Networks. http://www.techcrunch.com/2008/06/23/modeling-the-real-market-value-of-social-networks/, 2008.
2. R. Baden, A. Bender, D. Starin, N. Spring, and B. Bhattacharjee. Persona: An online social network with user-defined privacy. In *ACM SIGCOMM*, Barcelona, Spain, August 2009.
3. K. Bennett and C. Grotho. GAP - Practical Anonymous Networking. In *Privacy Enhancing Technologies*, pages 141–160, 2003.
4. L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks. In *18th Intl. World Wide Web Conference*, 2009.
5. S. Buchegger, D. Schiöberg, L. H. Vu, and A. Datta. PeerSoN: P2P Social Networking. In *Social Network Systems*, 2009.
6. I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In *Design Issues in Anonymity and Unobservability*, pages 46 – 66, 2000.
7. L.-A. Cutillo, R. Molva, and T. Strufe. Privacy preserving social networking through decentralization. In *IEEE WONS*, 2009.
8. L. A. Cutillo, R. Molva, and T. Strufe. Safebook : a privacy preserving online social network leveraging on real-life trust. 2009.
9. L.-A. Cutillo, R. Molva, and T. Strufe. Safebook: Feasibility of Transitive Cooperation for Privacy on a Decentralized Social Network. In *World of Wireless, Mobile and Multimedia Networks*, 2009.
10. S. Guha, N. Daswani, and R. Jain. An Experimental Study of the Skype Peer-to-Peer VoIP System. In *Peer-to-Peer Systems*.
11. S. Guha, K. Tang, and P. Francis. NOYB: Privacy in Online Social Networks. In *Online Social Networks*, pages 49–54, 2008.
12. T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, pages 94–100, 2007.
13. P. Maymounkov and D. Mazieres. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In *LNCS: P2P-Systems*, 2002.
14. S. Moyer and N. Hamiel. Satan is on My Friends List. http://www.blackhat.com/html/bh-usa-08/bh-usa-08-archive.html, 2008.
15. M. Rogers and S. Bhatti. How to Disappear Completely: A Survey of Private Peer-to-Peer Networks. 2007.
16. M. Steiner, D. Carra, and E. W. Biersack. Faster content access in KAD. In *Peer-to-Peer Computing*, Sep 2008.
17. C. M. A. Yeung, I. Liccardi, K. Lu, O. Seneviratne, and T. Berners-Lee. Decentralization: The Future of Online Social Networking. In *Future of Social Networking*, 2009.



(a) Profile data retrieval with delegation: user $\mathcal{U}$ retrieves $\mathcal{D}$'s reference from the DHT and delegates $\mathcal{V}$'s profile data request to his trusted contact $\mathcal{Z}$.



(b) Estimated RTT for data retrieval in 9 hops (first retrieval) and 5 hops (future retrieval).

Fig. 3. Profile data retrieval and estimated RTT assuming a successful DHT lookup in 4 hops, a matryoshka with 4 shells and request delegation.