Safebook: A Distributed Privacy Preserving Online Social Network

Leucio Antonio Cutillo Refik Molva Melek Önen
EURECOM
Sophia Antipolis, France
Email: {cutillo, molva, onen}@eurecom.fr

I. Introduction

Online Social Network (OSN) applications and services such as picture sharing, wall posting, and the like, nowadays have a strong impact on the way users interact with each other. Catering for a broad range of users of all ages, and a vast difference in social, educational, and national background, these applications and services allow even users with limited technical skills to share a wide range of personal information with a theoretically unlimited number of partners. This advantage comes at the cost of increased security and privacy exposures for users for two main reasons: first of all, users tend to disclose private personal information with little guard, and secondly, existing OSN applications severely suffer from vulnerabilities in their privacy protection or the lack thereof. The exploitation of these vulnerabilities [1] can lead a malicious user to launch many different types of attacks such as Id theft, profile cloning or secondary data collection [2]. Furthermore, even assuming a perfect protection from such malicious users, legitimate users are still exposed to a major orthogonal privacy threat, since in all existing OSN applications, the service provider has access to all the data including some private information stored and managed by the application itself and can misuse such information easily. Since the access to users' private data is the underpinning of a promising business model, current OSN services are not likely to address this problem in the near future. Researchers recently proposed to design the OSN application based on a peer-to-peer architecture in order to avoid centralized control over users' data. While in one hand a peer-to-peer model seems to be a good candidate to build a privacy preserving solution that avoids centralized control, on the other hand it lacks any a priori trust relationships among parties.

Among existing peer-to-peer based solutions, Safebook [3] leverages the social trust that is available as part of the very application in order to build a network of trusted peers that store OSN users' data. In Safebook each user's data is stored at the nodes of that user's trusted friends. The untraceability of the communications during look-up and data retrieval operations is assured thanks to an additional feature of Safebook in that the messages between a requester node and a friend's node that serves the request always route through several hops in order to hide a user's social links that are reflected by the OSN graph. Safebook defines two identities for each peer, namely the node and user identifiers, to prevent the disclosure of sensitive friendship information originating from an analysis on the data flows. Moreover, Safebook also prevents Sybil attacks thanks to the presence of a Trusted Identification Service (TIS) which is contacted only once during the user registration phase in order to generate a unique and unforgeable identifier per user. The introduction of this third party does not impact the decentralized nature of Safebook's architecture since it is not involved in any data communication or data management operation.

II. Overview of Safebook

In this section we briefly describe the core components of Safebook [3] and tackle the advantages in terms of privacy and security with respect to existing on-line social networking applications.

A. Safebook components

As mentioned in the introduction, in order to provide a privacy preserving and trusted OSN, as opposed to existing on-line social networking applications, Safebook adopts a decentralized architecture relying on the cooperation among peer-to-peer users while leveraging real-life social links. Instead of storing their private data at a single and centralized OSN provider, users replicate and store their content at several peers which are their real-life friends. Safebook features three main components that are illustrated in Figure 1:

- The first component of Safebook is the ***Matryoshka*** which is a mapping of the real social network graph: each user $\mathcal{V}$, namely the ***core***, is surrounded by concentric shells where the first shell regroups some of its trusted contacts (real-life friends) named as ***mirrors*** which are in charge of storing $\mathcal{V}$'s data. Starting from each mirror, each hop connects to a trusted node in order to construct a multi-hop communication link (***chain***) until the outermost shell is reached. Nodes in this last shell, called ***entrypoints***, act as a gateway for all data requests addressed to the core. A data request message reaches a node's mirror from an entrypoint through this path composed by hop-by-hop trust relationships. Thanks to the use of several layers in the

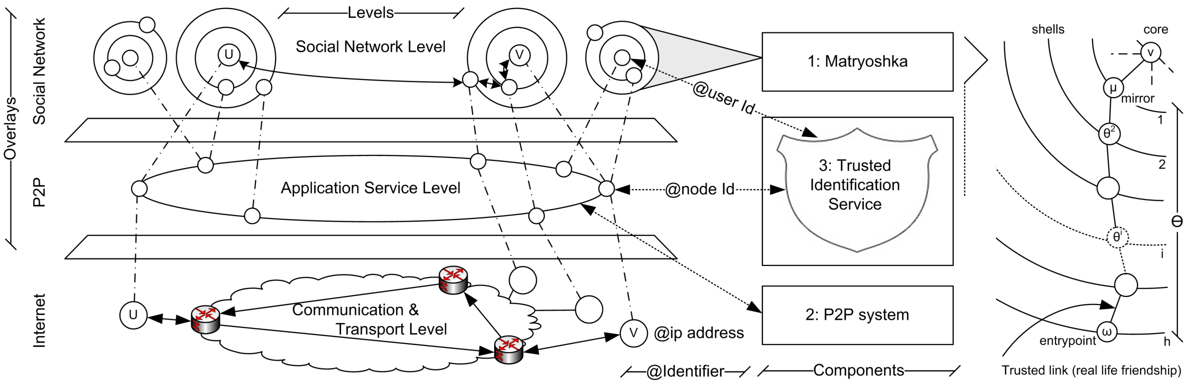


Figure 1. Safebook architecture(center) and Matryoshka graph (right)

Matryoshka, nodes requesting $\mathcal{V}$'s data cannot discover who its friends (the mirrors) are.

- The second component of Safebook is a P2P system used for profile data lookup, whose role is to provide the list of $\mathcal{V}$'s entrypoints to all the requesters $\mathcal{U}$.
- Finally, the third component of Safebook is the ***Trusted Identification System*** (***TIS***) and provides each node with a pair of unique ***node identifier*** and ***user identifier*** and the related certificates in order to prevent impersonation and in particular Sybil attacks. The TIS is not involved in any data management operation and does not need to be constantly on-line.

The matryoshkas and the P2P system constitute two different overlays on top of the internet. Currently, a DHT derived from KAD [4] is used as the P2P system.

III. Architecture of the Safebook prototype

The Safebook prototype[1] is written in python and can be executed on multiple operating systems such as Windows, Linux and MacOs. As depicted in figure 2, it is composed by four different managers:

1) the ***Communication Manager*** which is in charge of sending and receiving network packets;
2) the ***S2S Manager*** which mainly builds the DHT overlay;
3) the ***Matryoshka Manager***, which builds the social network overlay;
4) the ***User Manager*** which implements the user interface.

The Safebook client is an event-driven application: all managers send requests and responses to a dispatcher, and receive back indications or confirmations (internal messages). When two safebook clients communicate with each other, their respective communication managers send and receive network packets (external messages) (see fig.3).

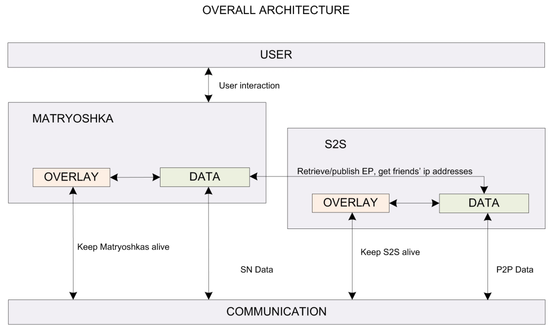


Figure 2. Overall architecture of Safebook.

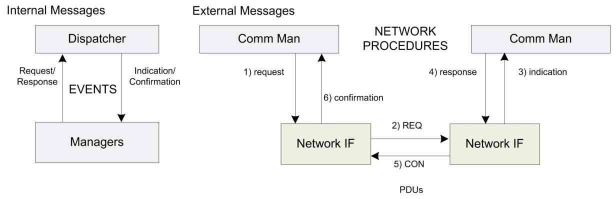


Figure 3. Internal (left) and external (right) message exchange in Safebook

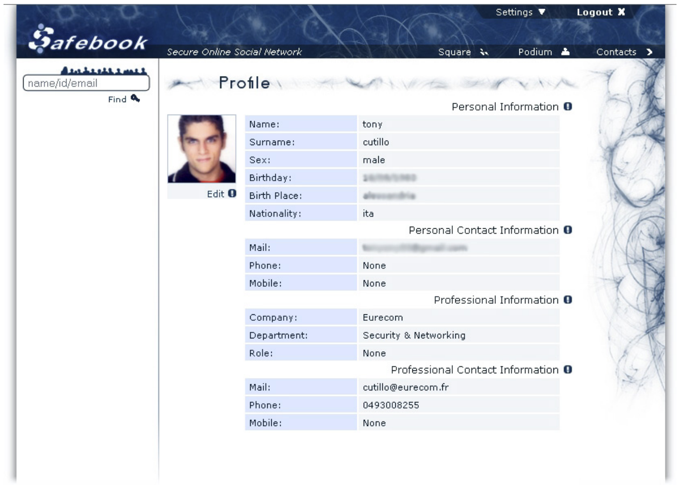
The user interface has been implemented under the form of a webpage (see fig.4), such as all current social network services which are accessible via internet browsers.

IV. Demonstration

In our demonstration, four key features of Safebook will be shown:

- joining the Safebook network: to join Safebook, a new member $\mathcal{U}$ provides the TIS with his identity property set, such as full name, birthdate, birthplace, gender etc., together with a proof of owning it. The TIS then computes and sends $\mathcal{U}$ his node and user identifiers together with a certificate associating a different public key generated by $\mathcal{U}$ himself for each identifier. $\mathcal{U}$ can

[1] http://www.safebook.us/prototype.html



Personal Information

Name: tony
Surname: cutillo
Sex: male
Birthday: [redacted]
Birth Place: [redacted]
Nationality: ita

Personal Contact Information

Mail: [redacted]
Phone: None
Mobile: None

Professional Information

Company: Eurecom
Department: Security & Networking
Role: None

Professional Contact Information

Mail: cutillo@eurecom.fr
Phone: 0493008255
Mobile: None

Figure 4. Graphical interface of Safebook.

then join the peerto-peer substrate based on his node identifier. $\mathcal{U}$ will also start his matryoshka registration process by creating a first chain thanks to a matryoshka bootstrapper node advertised by the TIS.

- friendship establishment: To add a user $\mathcal{V}$ as a friend, $\mathcal{U}$ first performs a lookup in the P2P overlay for the hash of a set of $\mathcal{V}$'s attributes, such as $\mathcal{V}$'s full name. This lookup is performed in a recursive way to hide the identity of the real requester, and returns the set of $\mathcal{V}$'s matryoshka entrypoints. At this point, $\mathcal{U}$ is able to send a friendship advertisement that is forwarded along $\mathcal{V}$'s matryoshka. Once notified about this friendship advertisement, $\mathcal{V}$ can accept the friendship and reply, in turn, with his friendship advertisement or simply discard it. In case of positive answer, $\mathcal{V}$'s friendship advertisement is forwarded along $\mathcal{U}$'s matryoshka.
- robustness: In case of successful friendship establishment, $\mathcal{U}$ tries to extend his matryoshka and build an additional multihop chain starting from $\mathcal{V}$. As a key aspect of Safebook, all the chains composing $\mathcal{U}$'s matryoshka are still alive even in case $\mathcal{U}$ goes offline. In case an intermediate node or an entrypoint log out, a chain is automatically rebuilt without requiring $\mathcal{U}$ to be online.
- data management:When a new chain is successfully built, $\mathcal{U}$ can use his friend $\mathcal{V}$ as a mirror, i.e. can store at $\mathcal{V}$'s place a replica of his own data. This data can be, at the moment, either $\mathcal{U}$'s profile or $\mathcal{U}$'s posts. To download $\mathcal{U}$'s data, a user $\mathcal{Z}$ performs a recursive lookup in the P2P overlay for an hash of $\mathcal{U}$'s attributes and receives $\mathcal{U}$'s current entrypoints. $\mathcal{Z}$ is then able to reach $\mathcal{U}$'s mirrors, no matter if $\mathcal{U}$ is online or not, and retrieve $\mathcal{U}$'s profile data. In case $\mathcal{U}$ leaves Safebook, when $\mathcal{U}$ logs in again, $\mathcal{U}$ tries to contacts his friends and to establish, from each of them, a matryoshka chain. In case a friend $\mathcal{V}$ is already serving $\mathcal{U}$, this request is immediately acknowledged.

Acknowledgment

This work has been supported by the SOCIALNETS project, grant agreement number 217141, funded by the EC seventh framework programme theme FP7-ICT-2007-8.2 for Pervasive Adaptation. See http://www.social-nets.eu/ for further details.

References

[1] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, "All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks," 2008, wWW 2009, Madrid.

[2] L. A. Cutillo, M. Manulis, and T. Strufe, "Privacy issues in online social networks," in *"Handbook of Social Network, Technologies and Applications", Springer, 2010*, 06 2010.

[3] 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, N.12, Consumer Communications and Networking Series, December 2009*, 2009.

[4] P. Maymounkov and D. Mazieres, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *P2P-Systems*, vol. 2429, 2002, pp. 53 – 65.