FIDIS

Future of Identity in the Information Society

Title:	“D13.1 Addendum: Identity and impact of privacy enhancing technologies”
Author:	WP13
Editors:	Dan Cvrcek (MU, Czech Republic)
	Vashek Matyas (MU, Czech Republic)
	Stefan Berthold (TUD, Germany)
Reviewer:	Jozef Vyskoc (VaF, Slovakia)
Identifier:	D13.1
Type:	[Deliverable report]
Version:	1.0
Date:	Monday, 31 March 2008
Status:	[Final]
Class:	[Public]
File:	wp13_1Add.doc

Summary

This document is an addendum to our report on technologies that enhance privacy from the technological point of view, and where we provided a review of technologies available.

The FIDIS NoE receives research funding from the Community’s Sixth Framework Program

Copyright Notice:

PLEASE NOTE: This document may change without notice – Updated versions of this document can be found at the FIDIS NoE website at www.fidis.net.

Members of the FIDIS consortium

1.	***Goethe University Frankfurt***	Germany
2.	***Joint Research Centre (JRC)***	Spain
3.	***Vrije Universiteit Brussel***	Belgium
4.	***Unabhängiges Landeszentrum für Datenschutz (ICPP)***	Germany
5.	***Institut Europeen D'Administration Des Affaires (INSEAD)***	France
6.	***University of Reading***	United Kingdom
7.	***Katholieke Universiteit Leuven***	Belgium
8.	***Tilburg University***[1]	Netherlands
9.	***Karlstads University***	Sweden
10.	***Technische Universität Berlin***	Germany
11.	***Technische Universität Dresden***	Germany
12.	***Albert-Ludwig-University Freiburg***	Germany
13.	***Masarykova universita v Brne (MU)***	Czech Republic
14.	***VaF Bratislava***	Slovakia
15.	***London School of Economics and Political Science (LSE)***	United Kingdom
16.	***Budapest University of Technology and Economics (ISTRI)***	Hungary
17.	***IBM Research GmbH***	Switzerland
18.	***Centre Technique de la Gendarmerie Nationale (CTGN)***	France
19.	***Netherlands Forensic Institute (NFI)***[2]	Netherlands
20.	***Virtual Identity and Privacy Research Center (VIP)***[3]	Switzerland
21.	***Europäisches Microsoft Innovations Center GmbH (EMIC)***	Germany
22.	***Institute of Communication and Computer Systems (ICCS)***	Greece
23.	***AXSionics AG***	Switzerland
24.	***SIRRIX AG Security Technologies***	Germany

[1] Legal name: Stichting Katholieke Universiteit Brabant

[2] Legal name: Ministerie Van Justitie

[3] Legal name: Berner Fachhochschule

Versions

Version	*Date*	*Description (Editor)*
0.3	13.02.2008	- First release (Dan Cvrcek)
0.4	6. 3. 2008	- Minor modifications, mail for internal review (Vashek Matyas)
0.5	28. 3. 2008	- Modification after internal review (Stefan Berthold, Vashek Matyas)
1.0	31. 3. 2008	- Final editing, submission (Vashek Matyas)

Foreword

FIDIS partners from various disciplines have contributed as authors to this document. The following list names the main contributors for the chapters of this document:

- Masarykova univerzita, Brno, Czech Republic (13)
- Technische Universitat Dresden, Germany (11)

Table of Contents

Executive Summary

This is an addendum to our report 13.1 following the comments from FIDIS review in the Summer 2007. This addendum reflects suggestions of the reviewers to add information about GNUnet, about the terminology issues and to clear some questions re. report conclusions.

The original report brings a comprehensive review of existing technologies enhancing privacy of users, and so it lays the cornerstone of FIDIS efforts to investigate the inter-relations of various aspects of identity as studied by FIDIS and of fundamental privacy issues, namely the impact of privacy enhancing technologies. These issues are also very closely related to profiling techniques as used for, e.g., traffic analysis.

A GNUnet

One of the technologies we have missed in the original text of the deliverable is GNUnet – it represents a technology for censorship-resistant file sharing. It is not a typical privacy enhancement technology, as the goal is not that much hide content of the data, or "author" of a piece of data, but rather place, where the given piece of data is stored. The attacker we are interested in here is someone trying to delete a file and make it unavailable. GNUnet [4, 3] is a "decentralized, anonymous, and censorship-resistant P2P framework" [2] and it enables users to request contents, for instance files, Web sites, or other data. Consequently, such contents will be delivered, if available within the network.

A.1 Introduction

GNUnet is a pure P2P system that is all users behave the same way with respect to use and supply of services and functionality. In particular, there is no (central) directory service as in Onion Routing or the Blender in Crowds [Deliverable 13.1, Section 7.2]. Contents available within the network are redundantly stored in a distributed manner on several clients. In general, however, none of the clients entirely stores entire file content, but a share of a file, for instance.

The most significant difference between GNUnet and similar organized systems like Crowds or Onion Routing is the objective. GNUnet intends to provide anonymity for requests, which target resources *within* the network rather than public web servers, for instance, which are outside the GNUnet. Thus, resources need to be explicitly propagated to the network. This difference is important for the preservation of anonymity, since a lot of known attacks, which affect other anonymity services, are based on linkability analysis of data which appears right on the border of such systems. That is, the adversary grasps such anonymity services as black boxes and watches their inputs and outputs. Connections can be worked out, for instance, by comparing the amount of data between users and anonymity service with the amount between anonymity service and a dedicated web server.

A.2 Base Layer

GNUnet is composed of several layers (see Figure 1) and provides itself a transport service which is connectionless and not reliable. The GNUnet base layer relies, in turn, on a transport service of the same quality, which is typically UDP. There are, however, also implementations utilising TCP, HTTP, or SMTP.

The base functionality of GNUnet is twofold and consists of (a) the exploration of new users and (b) the integrity-preserving, accountable, and confidential communication between users. For that reason, each GNUnet client generates a pair of RSA keys, which will be used as digital identity and for confidential communication simultaneously. The digital identity is propagated to other GNUnet clients while the new client registers.

In order to become part of the GNUnet network, a client first of all needs to know a subset of addresses belonging to clients who are already part of the network. In case of UDP as underlying transport layer, such addresses would be tuples, each consisting of an IP address and a port. The size of the address subset affects the speed of the registration process. The greater the subset is the faster the new client becomes known to the other clients. The actual process of registration is done by means of **HELO** messages. The new user then sends a **HELO** message to each of these addresses together with her own address, validity information, and her public RSA key. Additionally, the **HELO** message is signed by the client, using its secret RSA key.

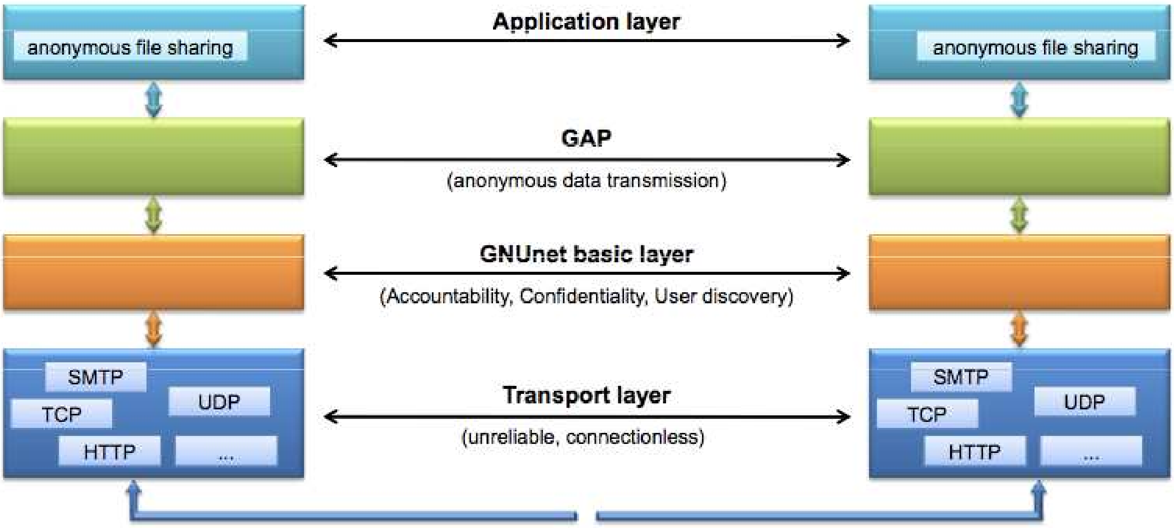

Figure 1: Layer model of GNUnet.

Thus, the new user proves that she has control over the secret key. In the next step, the new client tries to exchange a symmetric session key by means of asymmetric encryption with each client who received a HELO message. The session key is used for link encryption between adjoining clients. Additionally, **HELO** messages of new clients are distributed through the GNUnet network. That way, each client discovers more and more new clients by time. The distribution depends on the clients which received **HELO** messages. These clients support the distribution by forwarding received **HELO** messages to random clients. It is then up to the client, who receives new **HELO** messages, to decide whether an encrypted connection should be established to the originator of the **HELO** message or not.

A.3 Anonymity Layer

On top of this base layer lays the anonymity-preserving transport layer that is in terms of GNUnet terminology the *GNUnet's anonymity protocol* or *GAP*, in short. GAP is mainly useful for requesting files in an anonymous manner. By anonymity in terms of GAP we address the state in which an adversary is not able to prove (with likelihood greater than *p*) that a user is sender or recipient, respectively, of a message, which has been transmitted through the GNUnet network. This needs to hold even if the adversary is able to eavesdrop all connections within the network or able to alter transmitted data. Additionally, the anonymity must not be broken, if a set of members of the GNUnet network, which might be of almost arbitrary size, collaborates with the adversary, that is providing data to the adversary or be controlled by him.

Essentially, anonymity of a user is achieved by rerouting the requests and responses over different users of the GNUnet. In contrast to the proxy approach where anonymity for users is achieved by means of rerouting traffic over a third party, GNUnet is more similar to the

Crowds approach. Anonymity for a user in GNUnet is achieved by acting as relay for other users. Own messages can then be hidden in foreign traffic.

In order to achieve anonymity, that is p<1, a user needs to receive messages from at least one neighbour which does not collaborate with the adversary. In that case, the adversary cannot be sure, whether the data received from the user has been initiated by the user herself or is a rerouted data from another user. This is achieved due to encryption on GNUnet's base layer.

A.4 Data Processing

In GNUnet, files and data are transmitted in blocks B, where each block is of size 1024 Bytes. The content of each block is symmetrically encrypted, where the encryption key is the result of an application of the hash function h to the content of the block B, denoted as $h(B)$. In GNUnet, RIPEMD-160 [1] is used as hash function h. Thus the encrypted block E_{enc} , which is to be transmitted in GNUnet, is right $E_{enc} = E_{h(B)}(B)$. Encrypted blocks are stored in a hash map with lookup key $h(h(B))$. Lookup requests are of the form $h(h(h(B)))$.

Files or data that are larger than 1024 Bytes are to be distributed to several blocks. In addition, an index block $I = h(B_1), \ldots, h(B_{52}), CRC32(B_1, \ldots, B_{52})$ is created. This index block is then stored the same way as the content blocks, that is encrypted with key $h(I)$ in the same hash map with lookup key $h(h(I))$. In the case the data to be transmitted in GNUnet is greater than 52 Blocks, more index blocks will be created as needed. This yields a tree structure of index blocks. The root index is of particular importance, since it is encrypted with right that hash value of the lookup key, which is published in GNUnet for the entire block.

A.5 Request & Response

A main difference of GNUnet compared to Crowds or Onion Routing is the relation between request and response routes. In the other two systems, the routes are basically the same (just inverse and, thus, differ only in their direction). The difference of GNUnet is that both routes can be independently determined, to a certain degree (see Figure 2), by the user. In case of rerouting requests, a user A may decide, if she wants to set her address as originator of the message (indirecting) or if she wants to keep the address B of the original sender (forwarding). In the latter case, the response will be delivered straight to B without being rerouted over A.

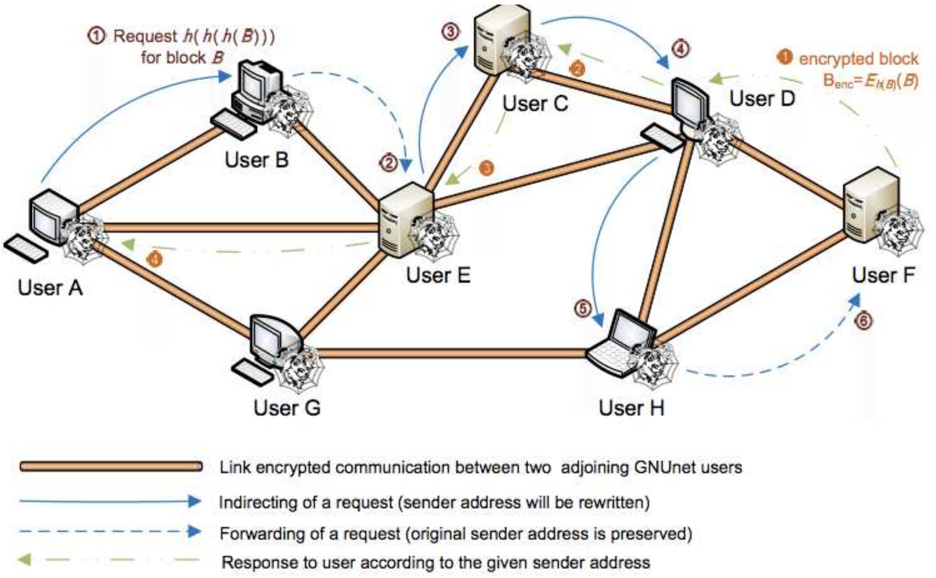
① Request $h(h(h(B)))$ for block B
③
④
❶ encrypted block $B_{enc}=E_{h(B)}(B)$
❷
User C
User D
User B
②
❸
User E
User A
❹
User F
⑤
⑥
User G
User H
Link encrypted communication between two adjoining GNUnet users
Indirecting of a request (sender address will be rewritten)
Forwarding of a request (original sender address is preserved)
Response to user according to the given sender address

Figure 2: Anonymous data transmission in GNUnet. User *A* sends a request to user *B* who is deciding to forward the request to user *E*. The response passes from user *F* to user *A* in only four steps. This is achieved by means of optional shortcuts.

It is up to each user to decide how many and which other users will receive a request which is going to be rerouted. The amount of forwarded messages is determined by the current CPU and network loads, the reputation of the sender, and a random value. In that case, reputation of a user is determined by the amount of her requests and valid answers, in short her behaviour. Additionally, each request provides a field, which determines the time-to-live that is particularly useful to avoid that requests circulate forever in the GNUnet.

Messages can be sent to the users in direct connection with the sender. However, for each message, which is going to be forwarded, the choice of recipients is not uniformly distributed, but rather depends on the hash value of their public keys. Recipients are the more preferred the more the public key's hash value is "close"[4] to the one of the request.

In general, requests (and responses) are not instantly delivered. It is rather the case that they will be buffered until either the buffer runs full or a random period of time exceeds.

It is then up to each user which receives responses to decide whether she wants to store the response or not. That way, requested data is distributed more and more within the GNUnet by time. Moreover, one user may ask another to store data. However, whether the other user complies depends on the reputation of the requester and local resources, particularly storage space.

A.6 References

[1] ISO: Information technology – Security techniques – Hash-functions – Part 3: Dedicated hash-functions, 2004. ISO Standard ISO/IEC 10118-3:2004.

[2] Website of the GNUnet project. September 2007. http://www.gnunet .org/.

[4] According to [3], "some metric" is used for calculating closeness.

[3] Krista Bennett and Christian Grothoff. GAP - practical anonymous networking. In *Proceedings of the Workshop on Privacy Enhancing Technologies 2003*, LNCS 2760, pages 141–160. Springer, Berlin, 2003.

[4] Christian Grothoff, Ioana Patrascu, Krista Bennett Tiberiu Stef, and Tzvetan Horozov. GNET, Whitepaper, June 2002. Version 0.5.2, http://www.gnunet.org/download/main.pdf

B Terminology

Andreas Pfitzmann from TU Dresden and Marit Hansen from ULD Kiel have undertaken a development of terminology for privacy related areas in 2000. There are many individuals contributing to the terminology, and majority of contributors to this FIDIS deliverable actually has been contributing to this terminology collection. The actual version (February 2008) is 0.31 and is available from http://dud.inf.tu-dresden.de/Anon_Terminology.shtml .

Although we have spent several pages on explaining some necessary terms in our deliverable, the referred document is the best available terminology and we recommend it to be used as the source of definitions for many terms we have been using throughout this deliverable. Also, the authors are open to comments and improvements and everyone is welcome to submit their opinions.

FIDIS is an interdisciplinary project and there have been several suggestions to create a dictionary or terminology document that would allow experts with different backgrounds understand each other. We believe that Pfitzmann's and Hansen's terminology is the best starting point as it tackles the problem from the technological point of view to a deep detail. Any acceptable dictionary between technologists, social scientists, lawyers, and other experts is the Grail and it is much more important to facilitate mutual understanding of these experts by providing truthful descriptions of the terms used in particular areas.

C Limits of Privacy Enhancing Technologies

The deliverable D13.1 lacks proper conclusions and they were omitted on purpose. It is an introductory document and the goal is to give an overview of the technologies and leave a lot of space for readers to think about particular technologies.

We have split privacy into two distinct parts – application and communication privacy. All the technologies used to provide privacy work very well, when used on random data and random communication. Unfortunately, humans are predictable and their behaviour features patterns that can be used to defeat or mitigate privacy properties offered by various technologies.

One can liken it to using strong cryptography with the same plain text all the time. It may be hard to decrypt the encrypted messages, but you soon realise that it is the same message being encrypted all over again.

There has been a lot of work done on quantifying privacy properties for communication anonymity systems, as described in Chapter 3, but we still miss verification of the results on large datasets of real data, real traffic. We believe that current estimates are reasonably good and there will be no changes in the order of magnitude, but the equations might get simplified or constants changed.

There is much less known about privacy in databases. There has been some research done in the area of medical databases, but we generally know much less about real properties of privacy enhancing technologies applied on databases or other types of stored data. One of the reasons is vagueness of threat model definitions and their relevancy to real-world systems.

We will present more detailed results about the technological limits of privacy enhancing technologies in the subsequent deliverables of our workpackage, including results obtained from analysis of large real-world datasets.