FIDIS

Future of Identity in the Information Society

Title:	Estimating Quality of Identities
Editors:	Stefan Berthold (TUD),
	Sandra Steinbrecher (TUD)
Reviewers:	David-Olivier Jaquet-Chiffelle (VIP),
	Daniel Cvrček (University of Cambridge)
Identifier:	D13.9
Type:	Deliverable
Version:	1.0
Date:	Tuesday, October 7, 2008
Status:	Final
Class:	Public
File:	fidis_d13.9.pdf

Summary

While in deliverable 13.8 the applicability of models/approaches for measuring privacy are illustrated by more-or-less declarative means, this deliverable focuses on testing and evaluating them. Due to the reason that real world data concerns real world people and their personal data, all data used were anonymised. Our main goals are demonstration of achievable results regarding privacy measurement by the data available for scientific research.

Copyright Notice:

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.	***Center 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)*
0.1	18.04.2008	Outline
0.2	30.06.2008	Incorporation of the MU contribution
0.3	03.07.2008	Incorporation of the KUL contribution
0.4	11.07.2008	Incorporation of the TUD contribution
0.5	11.07.2008	Incorporation of the ICPP contribution
0.6	29.07.2008	Editorial work and review by TUD
0.7	05.08.2008	Preparation for internal review
0.8	30.09.2008	Revision after internal review (all partners) Proofreading by TUD
0.9	02.10.2008	Additional figures from ICPP
1.0	07.10.2008	Final version

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:

Chapter	*Contributor(s)*
1 (Introduction)	Sandra Steinbrecher (TUD)
2 (Social Networks)	Claudia Díaz (KUL)
3 (User Profiling)	Marek Kumpošt and Vacláv (Vashek) Matyáš (MU), Stefan Berthold (TUD)
4 (Data Retention)	Martin Meints (ICPP), Stefan Köpsell and Stefan Berthold (TUD)
5 (Conclusions)	Sandra Steinbrecher (TUD)

Management Summary

The central aspect of FIDIS Workpackage 13 Privacy fundamentals is to deal with fundamental issues of privacy in relation to identity.

This deliverable must be seen in the light of a series of three deliverables within Workpackage 13, namely D13.6 *Privacy Modeling and Identity*, D13.8 *Applicability of privacy models*, and this deliverable D13.9 *Estimating quality of identity*.

Deliverable D13.6, provided a comprehensive insight into privacy (and context) modeling approaches, namely into technical and formal approaches to privacy quantification. Several different concepts were presented and discussed, based also on Deliverable D13.1.

Deliverable D13.8 focused on the applicability of privacy models and reviewed as well as illustrated the applicability of models from Deliverable D13.6 using real-world examples. But although the examples where from the real-world the applicability of models/approaches was still only illustrated by more-or-less declarative means, but did not consider real-world data. Our main goals are demonstration of (achievable) results by the data available for scientific research. The application areas we do apply our models to are social networks, profiling and data retention.

Table of Contents

1 Introduction

The central aspect of FIDIS Workpackage 13 Privacy fundamentals is to deal with fundamental issues of privacy in relation to identity.

This deliverable must be seen in the light of a series of three deliverables within Workpackage 13, namely D13.6 *Privacy Modeling and Identity*, D13.8 *Applicability of privacy models*, and this deliverable D13.9 *Estimating quality of identity*.

Deliverable D13.6 [KM07b], provided a comprehensive insight into privacy (and context) modeling approaches, namely into technical and formal approaches to privacy quantification. Several different concepts were presented and discussed, based also on Deliverable D13.1 [KM07a].

Deliverable D13.8 [JCAB08] focused on the applicability of privacy models and reviewed as well as illustrated the applicability of models from Deliverable D13.6 using real-world examples. But although the examples were from the real-world, the applicability of models/approaches was still only illustrated by more-or-less declarative means and did not consider real-world data.

For this reason in the present deliverable D13.9 we focus on testing and evaluating the models/approaches by real-world data. Due to the reason that real world data concern real world people and their personal data, all data used has been anonymised. But also both studies on anonymised data and the anonymised data itself to analyze it on our own were difficult to get. So we decided on collecting data ourselves in application areas we have under our control.

The application areas we do want to apply our models to are social networks, profiling and data retention:

We were able to do a study on profiling based on traffic collected at Masaryk university. Also we present an approach on profiling based on Netflix data by Narayanan and Shmatikov[NS08].

At Technische Universität Dresden, we operate an anonymity service AN.ON that could be used for tests on the effectiveness of data retention.

Unfortunately, for Social Networks no data could be collected in the time frame of the deliverable so we had to do some realistic simulations assuming typical structures of social networks.

2 Social Networks

2.1 Introduction

Social networks are social structures representable by graphs consisting of vertices (usually representing individuals) that are connected by edges (usually representing social relationships between the individuals).

Social network sites [BE07] are web-based services that allow individuals to

1. construct a public or semi-public profile within a bounded system
2. articulate a list of other users with whom they are connected, and
3. view and traverse their list of connections and those made by others within the system.

Thereby social networking sites offer and form virtual social networks. More and more Internet users become members of these virtual social networks because their real social network communicates mainly through this social networking site. Facebook[1] is a popular example. When Facebook started in 2004, the membership was restricted to students of colleges and universities but more than half of the 90 million users are outside colleges now [2].

The list of users a user is connected with in the social network usually is available to all other members. If he communicates with others, even if he does this anonymously, the one he communicates with is with very high probability also in the list of users he is connected with, usually his list of friends.

We consider the anonymity of users belonging to a social network who communicate with each other via anonymous messages. The attacker is the global passive adversary (she observes the inputs and outputs of the anonymous communication network) and also has knowledge of the users' profiles[3]. We first consider the two sources of information available to the adversary separately, and we combine them to examine what happens as the network grows. Interestingly, it turns out that the details of the mixing algorithm employed by the anonymous communication system play a significant role. Next, we briefly show how additional sources of information can be used by the attacker to further reduce anonymity. Finally, we look at how the quantity, quality and depth of knowledge about the users' relationships affects our results.

[1] `http://www.facebook.com/`

[2] `http://www.facebook.com/press/info.php?statistics` (last visit July 29th, 2008)

[3] In this chapter, we understand "user profiles" as a set of friends. Profiles will be discussed with a broader meaning in Chapter 3.

Our main contribution is evaluating how the uncertainty in the attacker's knowledge of user profiles affects anonymity. Indeed, we show that arbitrarily small errors in the profiles can lead to arbitrarily large errors in the anonymity probability distribution and hence point to the wrong subjects in the anonymity set. We develop the intuition behind this result and evaluate the errors in the anonymity probability distributions in the context of the social network. We conduct our experiments by simulation which helps us examine realistic scenarios.

2.2 Preliminaries

The system and attacker model considered in this chapter were already introduced in detail in the FIDIS Deliverable D13.8 [JCAB08], as well as the Bayesian methodology for combining the available sources of information. For convenience, we summarise here the most important points before proceeding with the analysis.

2.2.1 System and Attacker Model

We consider a system where a set U of N users send messages to each other through an anonymous communication channel modeled as a mix[4]. The adversary we consider is the *passive global adversary*, who can observe messages arriving and leaving the mix (as well their respective senders and recipients), but not its internal operations. Although the attacker does not know the correspondence between inputs and outputs, she is able to compute the probability distributions linking every input with all possible outputs and vice versa.

In addition to observing the mix inputs and outputs, the adversary has a priori knowledge of the users' sending behavior. We assume users to be linked via a social network, and that users send messages to those who are in their *profile*; i.e., their set of "friends." We have used various methods to generate the user sending profiles, which are described in detail [DTS08].

2.2.2 Anonymity with several sources of information

Let h_j be the hypothesis that user $u_j \in U$ is the sender (or recipient) of a given message received (or sent) by user $u \in U$, and $\Pr(h_j)$ the prior probability of this hypothesis being true. Let E be some evidence or observation that might give us additional information on the truthfulness of h_j, and $\Pr(E|h_j)$ be the probability of observing evidence E conditioned to h_j being true. Bayesian inference can be used to compute the posterior probability $\Pr(h_j|E)$

[4] Our analysis and experiments apply to any abstract anonymous communication channel for which probabilistic relationships between inputs and outputs can be derived.

of h_j, given that we have obtained evidence E. We denote this probability distribution by $P(H|E) = \{\Pr(h_j|E), 1 \leq j \leq N\}$:

$$\Pr(h_j|E) = \frac{\Pr(h_j)\Pr(E|h_j)}{\sum_{k=1}^{N}\Pr(h_k)\Pr(E|h_k)}$$

Bayesian inference can be applied recursively if new independent evidence E' becomes available to the adversary. We show results that introduce an additional source of information in Sect. 2.4.2.

2.3 Analysis

2.3.1 Intuition

The attackers' knowledge about the communication partners of users inside the social network comes from two sources—observing the mix and her a priori knowledge of the user profiles. Naturally, if we have a perfectly anonymous communication layer, the anonymity of the system comes only from the attacker's (lack of) information about the profiles. Conversely if the attacker has no information on the profiles of the users, she is restricted to observing the communications layer; i.e., the mix. The more complex setting when the attacker has knowledge of both is examined below.

Consider the case of users belonging to a vast social network and hence knowing a tiny fraction of the overall user population. In our model the attacker can see the inputs and the outputs of the mix and knows the profiles of all the users, so the only mixing that will take place is that between senders who share potential recipients or between recipients who share potential senders (note that the communication channel is modeled as a threshold mix, and thus, the chances that two senders who share common friends appearing in the same mixing round decrease with the number of users). Hence if the network grows and users' connectivity remains constant, anonymity falls due to the sparsity of the network. On the other hand, higher traffic load and number of users increase the anonymity provided by the mix. In Sect. 2.4.1 we show the tradeoff between these two effects.

The increasing popularity of blogs and, more generally, the availability of user-generated content makes it easy to gather a corpus of text linkable to an individual. Different people have different writing styles and patterns (such as word frequency or preferred grammatical constructions), and statistical tests that detect these patterns can be used to help identifying the authors of anonymous text. We study in Sect. 2.4.2 how the results of such a test can be combined with profiles and traffic analysis information, and its impact on sender anonymity.

The attacker's knowledge of the social network can vary in its quantity, quality and depth. She may know only of existence of links between individuals, the extent of those links, lack

knowledge of links in some part of the network and hence have to make do with approximations or, worst of all, assume wrong information. We assess the impact of each of these on anonymity in Sections 2.4.3, 2.4.4 and 2.4.5. Before proceeding to the results of the analysis, we give details of our experimental setup.

2.3.2 Experimental setup

We performed the analysis in the setting of a social network with a population of users arranged in a small-world network constructed following the Watts-Strogatz algorithm [WS98]. We also performed experiments on a scale free network [BA99] created with preferential attachment and the same number of average users, and the only noticeable difference was a larger variance in the results, which is due to the more uneven distribution of links per node in these networks. Unless indicated otherwise, we consider in our experiments 1000 users with an average of 20 friends each, arranged in a small world network with parameter $p = 0.1$ (i.e., highly clustered).

Users send messages only to their *friends* (i.e., users linked to them in the social network) with the probability specified in their profile. For the purposes of our experiments, we have developed several sets of user profiles with slightly different probability distributions. A detailed summary of the profiles used and the algorithms used to generate them can be found in [DTS08].

We chose a Mixmaster [Cot, UMS03] mix, as it is the most widely deployed high-latency network for anonymous email. The time intervals between users sending messages follow an exponential distribution with parameter λ, common to all users. We have chosen $1/\lambda$ to be 25 times greater than the timeout of the mix, so if users send messages on average once a day, the expected delay is between 30 min and 1 hour. In every experiment we simulate 130 rounds of mixing. We then extract the information which could have been observed by the attacker and compute the sender and receiver anonymity of each message.

2.4 Results

2.4.1 Growing the network

In this section we consider the anonymity of users as the social network is scaled up. To help develop the intuition we show the anonymity calculated from traffic analysis (mix input/output observations) and knowledge of the profiles separately. As the network grows, the anonymity provided by the mix increases as shown in Fig. 2.1(a)(Mix) simply because more traffic goes through it. As for the anonymity provided by the profiles (corresponding to Uniform profiles, cf. [DTS08]), we can see in Fig. 2.1(a)(Profile) that it remains constant, because we assume that the connectivity does not increase with the network (though in a real network it might increase slightly), which becomes more sparse. Interestingly, Fig. 2.1(a)(Combined) shows

that the combined anonymity decreases with the network size. As we shall see, variations in parameters that have a positive (mix) or no (profiles) effect on anonymity when sources of information are considered separately, can have a negative impact when all information is put together.

In this particular case the decrease in anonymity with network size is due to an interaction between profiles and mix function. Consider a random user Alice. The attacker is aware of her sender profile, so only users who share friends with Alice contribute to her anonymity. Alice and her friends send and receive on average the same number of messages whether the network is large or small. At the same time, the Mixmaster function [Cot] that determines the fraction f of messages sent per round increases with the traffic load until it reaches its limit[5]—note that in Fig. 2.1(a)(Combined) anonymity stabilises beyond that point. Therefore, the larger network induces the mix to flush a higher fraction of messages, which consequently in the mix for fewer rounds. This effect, in fact, decreases the amount of mixing, because friends of Alice who sent or received messages in the rounds before or after her contribute less to her anonymity[6].

2.4.2 Adding extra information

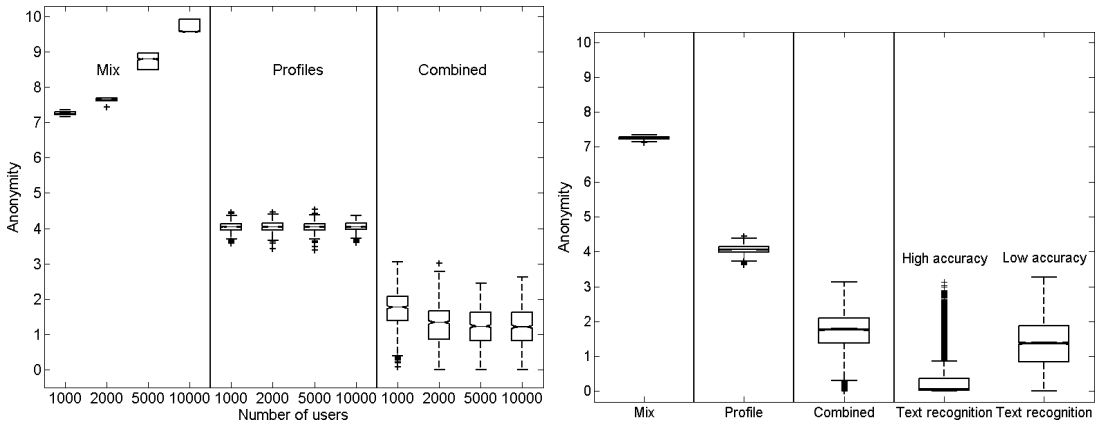
In this section we briefly show how Bayesian inference can be used to incorporate additional sources of information. Consider, for instance, a writing pattern recognition test. Let us assume that the attacker can run a test on the messages at the output of the mix and compare the writing to available text from the potential senders. The output of the test is 1 (with probability p_t) if the test considers that the user is the author, 0 otherwise (with probability p_f).

Based on the evidence E' produced by the test, the adversary can derive for each user $u_j \in U$ the probability $\Pr(h_j|E')$ that she was the true author of the text. Users testing negative (i.e., $u_j \in U_n$) have probability $\Pr(h_j|E' = 0)$ of being the writer, while those testing positive (i.e., $u_j \in U_p$) are the originator of the message with probability $\Pr(h_j|E' = 1)$.

The posterior probability distribution $P(H|E')$ is computed applying Bayesian inference as explained in Sect. 2.2.2. The evidence E' is a vector with zeros for users who tested negative and ones for those who tested positive, and we denote the j-th element as E'_j. The prior $P(H)$ corresponds to the (already existing) probability distribution that combines the profile and traffic analysis information. Assuming that E' contains k positives for a population of N

[5]The maximum fraction of messages sent by Mixmaster is $f = 0.65$. In our setting, this is reached when there are around 2500 users.

[6]friends who sent messages during the same round as Alice contribute the same amount as in the case of the smaller network



(a) Anonymity vs network growth

(b) Anonymity after combining three sources of information

Figure 2.1: Sender anonymity when various sources of information are available

users (i.e., $\sum_i E'_i = k$), $P(E'|H)$ is computed as follows:

$$\Pr(E'_j = 0 \wedge \sum_{i,i\neq j} E'_i = k|h_j) = (1-p_t)\binom{N-1}{k} p_f^k (1-p_f)^{N-k-1}$$

$$\Pr(E'_j = 1 \wedge \sum_{i,i\neq j} E'_i = k-1|h_j) = p_t\binom{N}{k-1} p_f^{k-1} (1-p_f)^{N-k}$$

We made experiments where we considered two tests that give correct answers with different degrees of accuracy. The high accuracy test had a true positive rate $p_t = 0.8$ and a false positive rate $p_f = 0.01$, while in the low accuracy one these values were $p_t = 0.5$ and $p_f = 0.1$. The results are shown in Figure 2.1(b), where we can see how the new information provided by the test reduces (on average) sender anonymity. Note however the outliers: in some instances, the additional information provided by text recognition test does not help reducing anonymity. We further investigate this effect in Sect. 2.4.6.

2.4.3 Quantity of profile knowledge

In the previous section we compared anonymity in these cases: (i) the adversary knows the profiles of all users, but cannot perform traffic analysis; (ii) the adversary does not know any profiles, but can observe the mix; and (iii) the adversary has access to all profiles and communication data. Here, we look at sender and recipient anonymity towards adversaries who can observe all traffic through the mix but only know a fraction of the user profiles (generated following the Uniform description; details about it are described in [DTS08]). We assume that the attacker has perfect knowledge of some profiles, and knows nothing about the rest. Whenever the attacker does not know a profile, she will consider it as uniform.

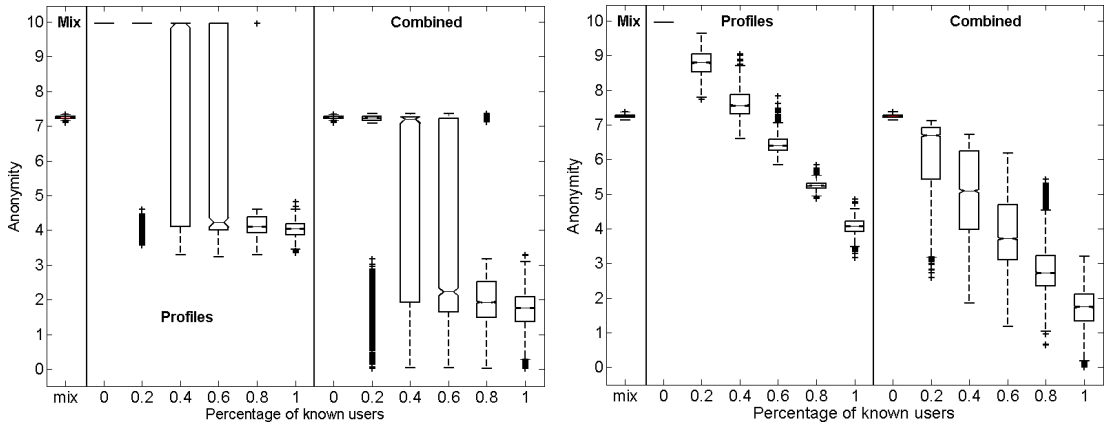


Figure 2.2: Receiver (left) and sender (right) anonymity depending on the quantity of profile knowledge

In Figure 2.2(a) we show the results for recipient anonymity with respect to the percentage of known profiles. On the left hand side of the figure (Mix) we show the anonymity A_m provided by the mix, which is independent from the quantity of profile knowledge and thus invariant for all experiments. The center and right hand side show, respectively, the anonymity A_p of the profiles and the combined A_c. Recipients of users with unknown profiles are unaffected by the percentage of known profiles, and enjoy the maximum anonymity that the mix can offer. For them, the profile anonymity is $A_p = log_2(N)$ and the combined recipient anonymity is $A_c = A_m$. Conversely, recipients of profiled users do not benefit from unknown profiles, and their recipient anonymity is the same regardless of how many other profiles are known. The aggregation of these two sets of recipient anonymity results can be clearly seen in the box plots of Fig. 2.2(a). Note the sudden jump in the median when half the profiles are known, and the values of the quartiles and outliers.

Unlike in the case of receiver anonymity, the percentage of known profiles affects the sender anonymity of all users, profiled or not, in the same way. This is because recipient profiles $P(u_i \leftarrow U)$ are computed using all sender profiles, and unpredictability of some users' sending patterns introduces uncertainty for all messages. The results of our experiments are shown in Fig 2.2(b)—as more profiles become available to the attacker, both the sender profile anonymity and the combined anonymity decrease.

Note that although the behaviour of sender and recipient anonymity is different when the adversary has partial knowledge, the values are the same for the extremes—i.e., sender and recipient anonymity are symmetric (in their distribution of values) both when all profiles are known and when all profiles are unknown, but not when some profiles are and some are not.

Finally, note that in our experiment all users have non-uniform sending profiles (they only send messages to their friends), so the adversary's assumption of uniform behaviour for unknown users introduces errors in her results. We further elaborate on the implications of having (or assuming) wrong information in the next section.

2.4.4 Quality of profile knowledge

Human behaviour is hard to model and predict, and even the most sophisticated adversary with access to vast amounts of information can only at best approximate user behavioural profiles. Therefore, we can reasonably assume that in a real world scenario there is going to be some difference between the profiles guessed or predicted by the adversary and the actual user sending patterns. Furthermore, due to the lack of available real-world data, little is known about how user sending profiles might actually look like, or how they evolve in time. For this reason, it is worth looking at the implications for the anonymity adversary of making wrong behavioural assumptions, such as assuming uniform sending profiles. In this section we study how noise in the profiles propagates and find that small errors in the profiles may lead to big errors in the end results.

There are many ways for the adversary to construct her guessed profiles. They can be obtained, to mention some examples, by studying the links between users in online social networks such as Facebook or LiveJournal, by analyzing user sending patterns when messages are sent over a non-anonymous channel (assuming that the user does not always use the mix for sending her messages), or by applying statistical disclosure attacks [DS04] to previous mix communications of the user. The profile construction method and the quality of data available to the adversary determine not only the accuracy of the profile, but also the nature of the "error" with respect to the real profile. For example, users may be linked in Facebook to acquaintances to whom they rarely or never send messages; they may have friends to whom they only communicate through an anonymous channel (and therefore do not appear in their non-anonymous communications); and the profiles obtained through disclosure attacks are noisy versions of the real sending patterns. Such a wide range of possibilities makes it hard to predict the type of profile errors we can expect in a real world scenario, and has led us to consider various kinds of erroneous profiles.

One important thing to note is the independence between error magnitude and actual anonymity value. Small errors in the final result indicate that the probability distribution obtained by the adversary is roughly similar to the one she would obtain had she used the true profiles; while large errors indicate that the adversary's view on who are the likely senders or receivers of a message is very different from the actual distribution computed with the real profiles—regardless of the entropy of the actual (guessed and true) distributions. Anonymity gives a measure of the adversary's uncertainty on who are the likely senders or recipients messages given that all available information is correct; while errors model the uncertainty of the adversary concerning the accuracy of her anonymity results, assuming that some information may not be correct.

In order to measure and compare the magnitude of the errors in the profile and final result making abstraction of the nature of the error, we use as metric the Euclidean distance $dist(x, y) = \sqrt{\sum_i (x(i) - y(i))^2}$ between true and guessed probability distributions. We have chosen Euclidean distance for its simplicity and well understood meaning, and because it provides clear bounds for the final error—the maximum distance between two probability

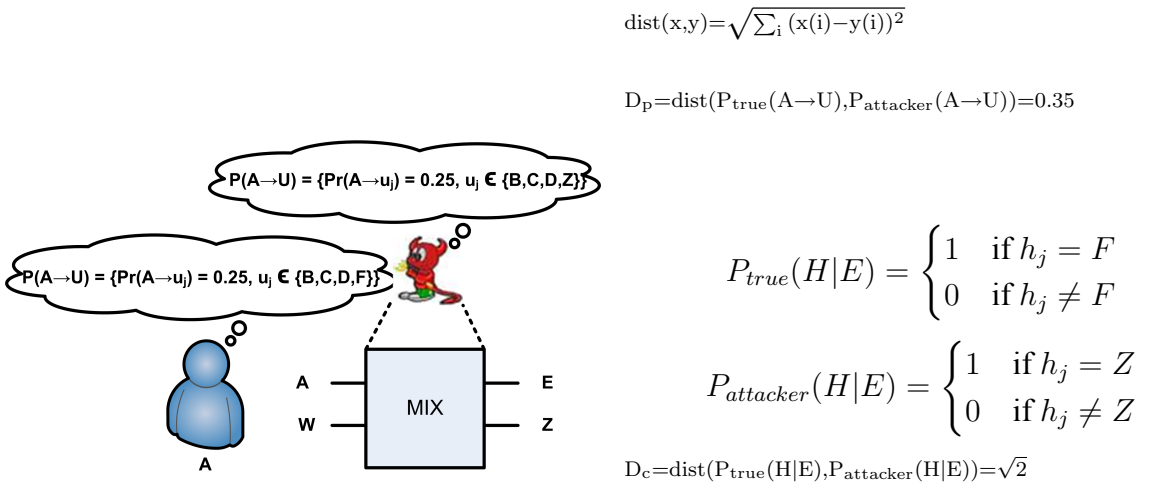
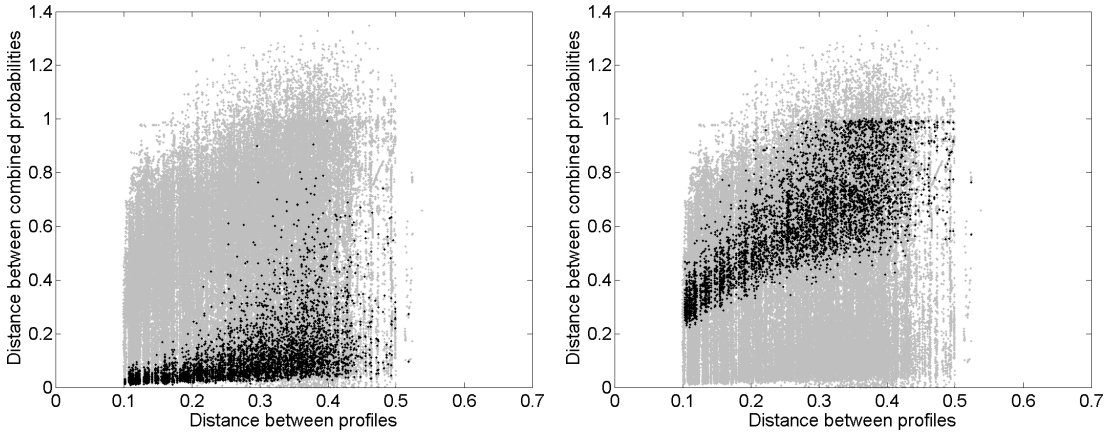


Figure 2.3: Example of how small errors in the profile can induce large errors in the attacker's results

distributions occurs only if they are orthogonal; its value is $\sqrt{2}$ and the minimum distance is 0.

Let us illustrate with a toy example our method for quantifying the impact of errors and the meaning of our results. Consider a simple scenario as the one depicted in Fig. 2.3, with a population $U = \{A, B, C, \ldots, Z\}$ and a unique (threshold) mixing round. User A sends with uniform probability $\Pr(A \rightarrow u_j) = 1/4$ to each of her four friends $\{B, C, D, F\}$, and with $\Pr(A \rightarrow u_j) = 0$ to the other users. The attacker, however, has a noisy version of A's profile, and believes that she chooses uniformly from the set $\{B, C, D, Z\}$. The attacker sees a single round of a threshold mix where A sends a message which comes out to either F or Z. Naturally, it was F as Z is not in A's true set of friends. The attacker, however believes it is Z, because he thinks that Z rather than F is in A's set of friends. Hence the wrong profile has led the attacker that Z is the recipient with probability one. We note that in this example, the receiver anonymity computed by the attacker when considering the wrong profile is zero ($A_{attacker} = 0$), as is the one she would obtain if she had precise knowledge of A's sending behavior ($A_{true} = 0$). However, the probability distribution obtained by the attacker is very different from the true result, and consequently her error is large. As the distance between the true and wrong results is much larger than the distance between the true and wrong profiles, this example provides the intuition that small errors in the profile may lead the attacker to completely wrong results.

Given that it is hard to predict the type of error the adversary is most likely to make, we have tested multiple instances of erroneous profiles. These include: *(i)* adding a *tail* to the profile distribution so that the probability of sending to non-friends appears greater than zero—yet significantly smaller than the one assigned to friends; *(ii)* introducing *Gaussian* noise; *(iii) eliminating* or *(iv) swapping* friends; and *(v)* assuming *uniform* behaviour. [DTS08] provides a detailed overview of the types of errors we have considered and the algorithms used to generated them.



(a) Error: tail (b) Error: uniform

Figure 2.4: Euclidean distance between true and guessed probability distributions vs distance between true and guessed profiles (quality of profile knowledge)

The results of our experiments are shown in Figures 2.4(a) and 2.4(b). In both figures, the X axis represents the distance between the true user profiles (with which the messages were generated) and the erroneous profiles considered by the attacker; i.e., $D_p = dist(P_{true}(A \rightarrow U), P_{attacker}(A \rightarrow U))$. The Y axis expresses the distance between the probability distributions the attacker would obtain with the correct and wrong profiles; i.e., $Dc = dist(P_{true}(H|E), P_{attacker}(H|E))$. The grey dots include results of experiments generated with the five error methods previously mentioned, and we have highlighted in black the results for two types of errors: adding a *tail* to the profile distribution (Fig. 2.4(a)) and assuming *uniform* profiles (Fig. 2.4(b)). We can see that the errors induced by adding a *tail* to the profile are relatively benign compared to other types in the background, as they take mostly low values in Y (note that this is the type of error obtained when learning users' profiles with a statistical disclosure attack). On the other hand, whenever the adversary (due to lack of information) assumes users send uniformly, she obtains a distribution that substantially deviates from the correct result—to the extent that she cannot have any confidence on whether or not she is getting a good approximation to the correct anonymity set. This is aggravated when we consider errors coming from swapping or eliminating friends, which cover most of grey area.

2.4.5 Depth of profile knowledge

In some practical scenarios (e.g., Facebook) the adversary may guess the friendship graph but lack enough data to estimate the strength of links between friends. We say that the adversary's guessed profiles lack *depth* when she cannot estimate the frequency with which friends are chosen as recipients, in spite of accurately distinguishing friends from non-friends (to whom users never send messages). In these circumstances, the best the adversary can do is to consider that recipients are picked uniformly at random from the set of friends. This is a special case of erroneous profiles like those analyzed in the previous section, but we have chosen to present

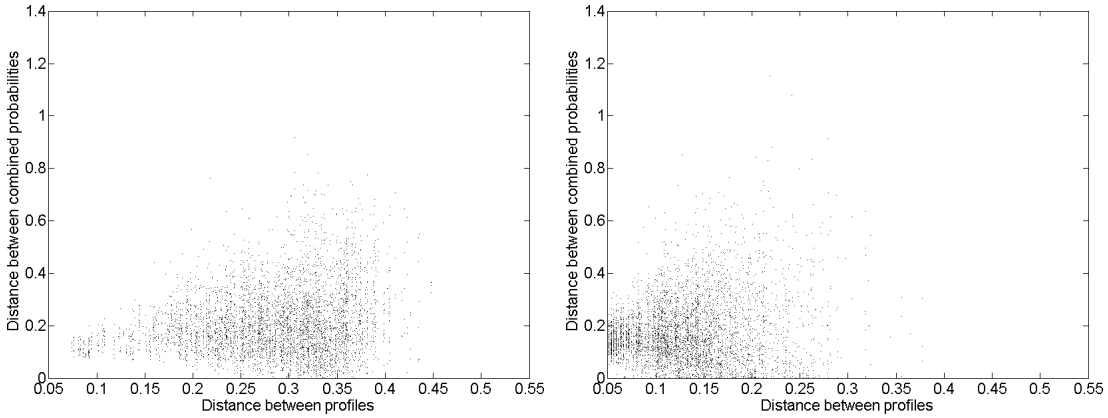


Figure 2.5: Receiver (left) and sender (right) anonymity error depending on depth of knowledge

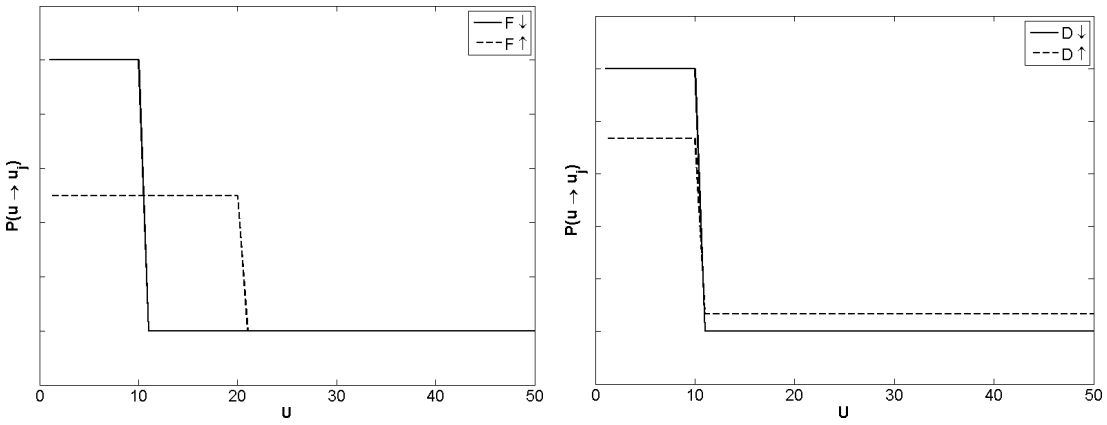
it separately for two reasons: first, because of its practical relevance (such profiles would be reasonably easy to construct); and second, although the profiles are noisy, correctly identifying friends (and non-friends) already provides very valuable information to the attacker.

To better illustrate the impact of the attacker's assumption, we consider that users choose their partners of communication having strong preferences for some of them (Skewed in [DTS08]). In Fig 2.5 we show how the error in the combined probability increases proportionally to the error in the profile. When the true profile of a user is close to uniform[7], the assumption of the attacker is not far from the truth—the distance D_p between both profiles is small, and so the distance D_c between the combined distributions. As D_p increases, so does D_c, but as a rule of thumb we could say that the error D_c is most likely to be smaller than the original error D_p. The contrast with the previous section's results (considering profiles uniform in the whole population) indicates that an adversary who correctly identifies friendship links obtains two advantages: she eliminates non-friends from the anonymity sets, effectively decreasing anonymity; and she has higher confidence in her result, because the true and guessed distributions are comparatively closer to each other.

2.4.6 How often does additional information reduce uncertainty?

It was pointed out in [DTD07] that in some cases additional information may result in higher anonymity, even if on average anonymity decreases as more information becomes available. In this section we present some results showing under which conditions we can expect these cases to appear. In all experiments we used Mixmaster (i.e., the anonymity A_m provided by the mix is invariant), and a small world network with 1000 users that send to friends with probability $\Pr_f$ and to non-friends with $\Pr_{nf}$, such that $0 \leq \Pr_{nf} \leq \Pr_f$. The details

[7] Because of the algorithm used to generate the profiles, cf. [DTS08], recipient profiles are on average more uniform than sender profiles, this explains why the values in Fig 2.5(b) are smaller than in Fig 2.5(a).



(a) Fixed D and variable F (b) Fixed F and variable D

Figure 2.6: Variation of profiles with F and D

of the generation of profiles are as follows: users with these profiles send messages to the whole population. Nevertheless, they choose their friends as recipients more frequently than non-friends. For user u, the probability assigned in her profile to each of her friends u_f is $\Pr_f = \frac{1/F+D/N}{1+D}$, while the probability assigned to each non-friend u_{nf} is $\Pr_{nf} = \frac{D/N}{1+D}$. F is the cardinality of the set of friends, and the influence of its variation in the profile can be seen in Fig. 2.6(a). The parameter D influences the relation, in terms of probability, between friends and non-friends. As D increases, the sending profile becomes more uniform in all N potential recipients, diminishing the difference between friends and non-friends, as shown in Fig. 2.6(b). For $D = 0$, users never send to non-friends, and profiles are uniform on the whole population for $D = \infty$.

We study the results according to two variables: the number F of friends per user, and a parameter $0 \leq D \leq \infty$ that tunes the difference between $\Pr_f$ and $\Pr_{nf}$, such that $D = 0$ implies $\Pr_{nf} = 0$, and $D = \infty$ implies $\Pr_{nf} = \Pr_f$.

To better understand how sending behaviour affects anonymity, we have studied separately the frequency of cases where the combined anonymity A_c is higher than the anonymity of the mix alone A_m or the profile A_p, and its variation with the parameters F and D. The results in Figs. 2.7(a) and 2.7(b) show, respectively, the percentages of messages for which $A_c > A_m$ and $A_c > A_p$, which we denote $f_{c>m}$ and $f_{c>p}$.

To interpret the results, note that increasing F and/or D leaves A_m constant; increases A_p (because it makes the profile more uniform); and A_c increases as well as a result of more uniform profiles. When $D = 0$ users *only* send to friends—i.e., the recipient anonymity set is reduced drastically—and A_c is always lower than A_m and A_p. For $0 < D < 1$ and small F, A_p has increased only slightly, while A_c benefits mostly from messages sent to non-friends—these are "rare[8] events" in which the hints coming from the mix and the profile are "contradictory."

[8]Note that for $D = 1$ half the messages are sent to non-friends, even if the probability of picking a concrete non-friend is small.

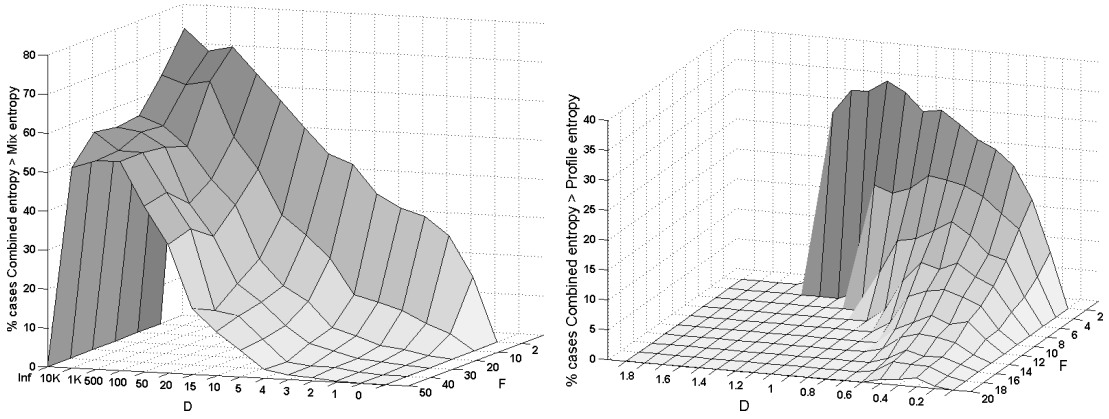


Figure 2.7: Percentage of cases where the combined anonymity is higher than the anonymity of the mix only $f_{c>m}$ (a) and profile only $f_{c>p}$(b)

Given the profile always points to the highest probability friends, when the mix points to (less probable) non-friends as most likely recipients, the mix and profile distributions compensate instead of reinforcing each other, making the combined distribution more uniform than one or both originals—i.e., $A_c > A_p$ and/or $A_c > A_m$. This also explains the high $f_{c>m}$ for larger values of D. Once F and/or D grow to make $A_p > A_m$, it becomes harder for A_c to catch up with it (we can see in the Fig. 2.7(b) that $f_{c>p} = 0$ for $F > 15$ and/or $D > 1.25$). When A_p hits its maximum with *perfectly* uniform profiles at $D = \infty$, the profiles stop bringing any additional information and $A_c = A_m$. Thus, $f_{c>m} = 0$ at $D = \infty$ in Fig. 2.7(a).

2.5 Conclusions and future work

In this section we examined the anonymity of users in the practical context of social networks. We showed the overall anonymity is low and likely does not increase with the size of the social network—if anything, it decreases as the network becomes more sparse.

The positive result of this work is that it is necessary to trust the social network entirely to provide high quality information about the sender profiles of the users, otherwise big mistakes can be made in the sender and receiver anonymity of messages. Indeed, unless the profile is perfect, the results may be meaningless as we demonstrated occurrences of huge errors in the anonymity probability distribution even when the profile error is small. We have found however that certain types of errors induce more bounded deviations than others in the overall anonymity.

Many issues remain to be addressed, particularly in the practical setting. Particularly interesting to us is the problem of assessing the anonymity of a real social network such as Facebook and its approximation as mapped by the attacker. Although we believe that we modeled the "friendship" between users to a fair degree of accuracy by using a Watts-Strogatz graph, the extent of the linkage and the resulting sender profiles remain a more difficult issue. Only

empirical modeling can gauge how much the real social dynamics differ from the theoretical models employed here.

One extremely promising line of research is to set up and evaluate an attack where the adversary continuously updates the social network graph with new information gained from observing the communication patterns and simultaneously tries to deanonymise the messages. Interestingly, our result holds in this setting too – whatever the methodology of deriving the social network graph, small errors in the graph may cause large errors in the anonymity of the message. Although complex statistical disclosure attacks may prove efficient at minimising the errors in the graph, they can never eliminate such inaccuracies which may arise as a result of external factors, for instance changes of user behaviour over time.

3 User Profiling

The research on profiling was described within FIDIS WP 7, especially Deliverable 7.2 [HB05]: "Profiling can provisionally be described as the process of constructing or applying a profile of an individual or a group. This process involves techniques (methods) and technologies (combination of tangible instruments and techniques; hardware and software). Apart from being a technique and a technology, profiling is also a practice: a specific way of doing things, within specific contexts, with specific purposes. It requires a learning process that integrates explicit with implicit knowledge. This means that profiling is a matter of expertise, of professional training and involves more than the mechanical application of explicit rules and procedures."

In this deliverable we concentrate on profiling as a technique commonly used in services (typically online) that want to "know more" about their users to provide customizable content (e.g. more appropriate search results, goods covering users' preferences, . . .). The question is what kind of information this customization might be based on. Some evidence about users is needed to provide personalised content (e.g. a list of previously searched terms, previous activity with the systems, etc.). This is typically called a *user's profile* and is based on their past activity. Instead of requiring users to provide this evidence, an online service could learn relevant *context* information itself.

3.1 Large Amount of Data and User Profiles: Case of University-Wide Network Analysis

In this section we develop a model that represents users' behaviour based on their past activity (i.e. based on all available context information) and evaluate its expressing value (in terms of its ability to pinpoint users among others based only on behavioural characteristics). Accuracy of the model is a key feature once it is used for reasoning about future actions that a user may likely perform.

One of the main questions we address is the quantification of how much private information can be derived from context information that is related to an individual [MC04] and to what extent a behaviour pattern can be used to pinpoint an individual among others. We will study various ways behavioural patterns can be built as well as the influence of different conditions (in terms of input data used) on the quality and stability of the patterns (user profiles).

This task involves working with relatively huge amount of input data. The reason is clear: the more input data we have the more precise profiles we can create. But discovering valuable

information in huge databases typically involves several phases such as data pre-processing and cleaning; data transformation into some common structure; data mining techniques to identify interesting patterns or correlations among parts of input data; final profiling and results interpretation; and evaluations under different conditions with different amounts and types of input data.

This work follows-up previous deliverable D13.8 [JCAB08] where an overview of practical application of modeling approaches and the necessary data preparation was provided. For this reason will only slightly touch the process of data preparation in section 3.1.1. We introduce the dataset we use for all our experiments and discuss the pre-processing phase to decrease the amount (in terms of removing "noisy" information) of information we will use in subsequent steps.

In section 3.1.2 we provide a detailed description of the profiling process and the similarity measure with all relevant aspects and some illustrative results.

In section 3.1.3 we discuss the way how the proposed similarity measure can be evaluated in terms of its accuracy. Our focus lies on describing the profiling power under different initial conditions. We discuss the influence of Inverse Document Frequency (IDF) as a tool that helps the model to be more accurate, application of thresholds on the similarity index and their influence.

3.1.1 Input data and its preparation

As input data we use global IP traffic log (Netflow) of Masaryk university network that collects information about all IP traffic going outside the network (we have agreed not to disclose any data from this log so all source and destination IP addresses are represented as capital letters or numbers). This data collection is extremely large due to high everyday load (around 3 million records every day), which is good from the profiling perspective. On the other hand, the need to restrict the information we will take into account is obvious. Optimised data forms vectors (later on we will describe how the optimisation works) of behaviour that are used as the input for the profiling.

This collection of data can be considered more or less as "low-level" information about the users since it provides information about communication flows only. The real content is hidden and therefore any kind of profiling leading to e.g. what the user feels or what is the user's current thinking is not possible.

Source IP addresses, destination IP addresses, destination port and the number of connections that were made during a particular period of time (day(s), week(s) or month(s)) is important information from the database. Based on this information, we can create a two-dimensional matrix, where each column is one destination IP address and each row is one source IP address. Every cell (i, j) then contains the number of connections that were initiated from the source IP address i to the destination IP address j. Each row of this matrix reflects the communication pattern of a particular source IP address (we will call it a vector of source IP address

behaviour). Problems with this matrix are its large dimensions and presence of many zeroes. Optimisations therefore aim to decrease both the size and the number of zeroes in the matrix.

We deployed several special techniques to optimise the matrix. Before we start describing them in more detail, we should mention the very first set of basic restrictions applied. In this step we create a new database table that consists of selected source IP address ranges and destination ports (e.g., just one faculty and ports 22, 80 – ssh, http). This helps focusing on data that are expected to have some inner relations. We cannot expect great similarities across multiple departments, because of the different interests of people working there. An example of a good type of restriction can be an IP address range of a particular department and a network-covered dormitory where students of this department live. We can expect that students will tend to exhibit similar activity no matter where they get connected to the network. But still, with this type of specification, the size of the resulting matrix is quite big so we have to go further with the optimisations.

The first idea in source IP address optimisation is that there are several levels of source IP address "activity" – some sources are quite passive in communication, while other sources are extremely active. If there is a set of IP addresses with similar behaviour, it will most likely exhibit some similarities in frequency of communication as well. Therefore separating source IP addresses into different levels of activity may help in increasing the accuracy of the consecutive processes [Kum07].

For every source IP address, we build a histogram describing how many destinations that machine on the given IP address visited how many times during a fixed period of time (e.g. one month). Information from each histogram is considered as a vector that reflects the frequency behaviour of each source IP address. The set of these vectors is processed with a clustering algorithm to find clusters consisting of IP addresses with similar frequency tendencies. As a clustering method we use Ward's clustering technique [War63] which (in our experiments) has a better tendency to produce quite sharp and balanced clusters (e.g. when compared with the complete linkage). This approach seems effective to reduce the size of the matrix with regard to the source IP address and let us concentrate to sources that have already exposed some similar tendencies.

The number of destinations depends heavily on the restrictions applied to the set of source IP addresses. Therefore this optimisation should be applied only if needed (contrary to the source IPs' addresses optimisation, use of which seems reasonable under any circumstances).

To optimise destinations we use technique that is commonly used in the area of text mining and information retrieval. It is called TF-IDF (Term Frequency - Inverse Document Frequency) and is a metric that evaluates relevancy of a word in a collection of documents [Ber03, LB04]. The formula is:

$$weight(i,j) = tf_{i,j} \cdot log_2(n/df_i), \text{ if } tf_{i,j} \geq 1 \tag{3.1}$$

where $tf_{i,j}$ is the number of occurrences of the i-th term within the j-th document d_j and df_i is the number of documents (out of n) in which the term appears.

If we consider one destination IP address as a term and a vector that describes source IP address behaviour as a document, we can use TF-IDF to evaluate the importance of a given destination for a given source in a collection of all sources and destinations. If we do not use TF-IDF, but only IDF ($log_2(n/df_i)$) we can generalise the approach to evaluate how important a destination is for a given set of sources. This is precisely what we are interested in. We can also consider this type of information as a context information since it highly depends on given sets of both source and destination IP addresses. This "index of importance" (for every destination IP address) will vary with every little change of either set of IP addresses.

Boundaries within which IDF falls range from zero (this is the case where every source visited that destination – $log_2(n/n)$) to $log_2(n/1)$ (this is the case where only one source visited that destination).

We use the IDF weight later in the similarity measurement to "adjust" behavioural vectors in a way that relevant components are strengthened and vice versa. We will describe this approach in the following section.

3.1.2 Behavioural vectors and user profiling

Once we apply some of the restrictions described in the previous section, we obtain a matrix of behavioural vectors. Rows of the matrix describe behaviour of each source IP address and form a vector. These vectors can be processed using cluster analysis (in general any technique to determine similarities can be applied) to find sets of source IP addresses that tend to have similar behaviour (with respect to the restrictions we applied).

In this section we will discuss the method we use to calculate similarities of behavioural vectors and provide some illustrative results. Questions we have to answer are what form of similarity measure we should use and how to express the "degree" of similarity. Since we have behavioural characteristics represented as vectors the similarity measure should reflect this fact. Our review of possible approaches ended up with cosine similarity measure which is very widely used in the area of text mining (information retrieval) and works with vectors. Cosine similarity measures the angle between two vectors, so it perfectly fulfils our second requirement – expressing the "degree" of similarity. The formula for cosine similarity between two vectors of the same dimensions $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots, b_n)$ is:

$$\cos(A, B) = \frac{A \cdot B}{|A||B|}, \tag{3.2}$$

where $A \cdot B$ is

$$A \cdot B = \sum_{i=1}^{n} a_i b_i, \tag{3.3}$$

and $|A|$ is a size of vector A

$$|A| = \sqrt{\sum_{i=1}^{n} a_i^2}. \tag{3.4}$$

So, finally

$$\cos(A, B) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \sqrt{\sum_{i=1}^{n} b_i^2}}. \tag{3.5}$$

Output of$\cos$ ranges from 0 (vectors are completely unrelated; we have only positive numbers or zeroes in all vectors) to 1 (vectors are completely related). So this number will indicate how similar given vectors are. The use of this metric is straightforward and gives reasonable results (see Tables 3.1 and 3.2). Given some information about destination IP addresses (from the optimisation phase) we considered whether any of these can also be incorporated in the process of similarity calculation. IDF is quite good to calculate actual relevancy for a given set of sources and destinations so we tried to utilise this information here again. Having a vector of behaviour it is obvious that some components are more relevant with respect to the actual set of source IP addresses, while some of them not (for example Google is very popular and almost everyone visits this destination, so the relevancy would be much lower than some exotic destination). IDF index ranges from 1 (the case a given destination is not very relevant with respect to the set of sources) to $log_2(n/1)$, which is the case a given destination is highly relevant with respect to the given set of sources (well, in the case of $log_2(n/1)$ only one source actually visited that destination). If we calculate IDF values for all destinations we obtain a vector which size is the same as for behavioural vectors. Each component of IDF vector represents actual relevancy of corresponding destination IP address. We use this IDF vector to balance behavioural vectors with relevancies – multiplication of all components with respective IDF values. This helps to emphasise components in behavioural vectors that are of greater relevance. Having this approach applied, vectors that are highly similar will be put even closer (compared to $\cos$ calculation without IDF) and vectors whose similarity is low will be put even more far away (compared to$\cos$ calculation without IDF). So the most important observation is that the use of IDF to balance behavioural vectors leads to more precise results in terms of similarity (see Section 3.1.3).

Technical aspects of this approach are quite straightforward.

1. Apply some restrictions and build a matrix of vectors we store this into a database table (we may consider this one as a training set).
2. Calculate IDF values for every destination IP address and put them into a database.
3. Apply the same set of restrictions but take different period of time (e.g. a month) and again store vectors into a database table (we may consider this one as a testing set).

Having all three tables filled with data we have to synchronise them in order to have vectors of the same dimensions.

At this point we further explore two promising approaches to set the initial restrictions.

- In the first case, we require only those source IP addresses that performed at least n communications and the same rule is applied on the second (testing) set.

A	:	1(A, 1), 1(B, 1), 1(O, 1), 0.164399(E, 1)
B	:	1(A, 1), 1(B, 1), 1(O, 1), 0.164399(E, 1)
D	:	0.999635(M, 1), 0.997976(D, 2), 0.0270172(J, 1)
E	:	0.999168(E, 2), 0.124035(A, 1), 0.124035(B, 1), 0.124035(O, 1)
J	:	1(J, 1), 0.0905358(D, 1)

Table 3.1: Cosine similarity values without using the Inverse Document Frequency. The number before brackets indicates the similarity index; capital letter indicates the similar source IP address and the last information is the number of common destination IP addresses.

A	:	1(A, 1), 1(B, 1), 1(O, 1), 0.0763637(E, 1)
B	:	1(A, 1), 1(B, 1), 1(O, 1), 0.0763637(E, 1)
D	:	0.999806(M, 1), 0.998918(D, 2), 0.0197195(J, 1)
E	:	0.999818(E, 2), 0.0573459(A, 1), 0.0573459(B, 1) 0.0573459(O, 1)
J	:	1(J, 1), 0.0661965(D, 1)

Table 3.2: Cosine similarity values influenced by the Inverse Document Frequency. The number before brackets indicates the similarity index; capital letter indicates the similar source IP address and the last information is the number of common destination IP addresses.

- In the second case the minimal number of communications is the same rule applied, but than we take the list of destination IP addresses from the result and use it as a restriction to prepare the testing set.

We provide an evaluation of either approach in the Section 3.1.3.

The calculation itself is done in a way that it is firstly computed using just the$\cos$ metric and secondly using$\cos$ influenced by the IDF vector. For each source IP address from the training set we calculate similarities with all source IP addresses from the testing set. Results (i.e. similarity value with/without IDF, number of common destination IP addresses) are stored in a database table.

There are four tables with some basic results from the similarity measurement process. Table 3.1 corresponds to$\cos$ values only (without the influence of IDF values) while Table 3.2 reflects IDF to optimise behavioural vectors. After a capital letter (a source IP address from the training data set) there is a list of most likely similar source IP addresses from the testing data set. The number before brackets states the similarity value; the capital letter in the brackets the similar source IP address and the last number before the closing bracket indicates the number of common destination IP addresses.

	1	2	3	4	5	6	7	8	9
A	0	0	0	0	0	0	0	0	1853
B	0	0	0	0	0	0	0	0	297
D	0	0	37	0	0	0	1	0	0
E	0	0	0	0	32	0	0	0	4
J	0	0	0	0	0	0	17	0	0

Table 3.3: Table of behavioural vectors (training set) – rows represent behavioural vectors for source IP addresses, columns are destination IP addresses.

	1	2	3	4	5	6	7	8	9
A	0	0	0	0	0	0	0	0	1487
B	0	0	0	0	0	0	0	0	244
E	0	0	0	0	12	0	0	0	2
J	0	0	0	0	0	0	12	0	0
D	0	0	11	0	0	0	1	0	0
M	0	0	5	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	3

Table 3.4: Table of behavioural vectors (testing set) – rows represent behavioural vectors for source IP addresses, columns are destination IP addresses.

So for example for source address D the list of most similar IP addresses contains M and D. We can see that the similarity values are quite close to each other. This is because the IDF value for the third destination is higher than for the seventh one. Behavioural vectors of all involved source addresses can be found in Tables 3.3 and 3.4.

Tables 3.3 and 3.4 contain behavioural vectors for all source (capital letters) and destination (numbers) IP addresses involved in the similarity measurement process. Numbers in tables indicate the amount of communication that the set of source IP addresses did. Table 3.3 may be referred as to a “training data set” while the Table 3.4 is used as a “testing data set”. In accordance to the agreement with our IT department all real source and destination IP addresses were covered up.

That was the description of the similarity measurement process supported by a set of results and explanations. At this point we are interested in how the model performs with different input data as well as conditions for calculating similarities (the influence of IDF, the amount of input data, initial input data optimisations, …). In order to answer these questions we need to do an evaluation with many tables calculating similarities based on different initial conditions. Description of the whole process is in the following section.

3.1.3 Evaluation of the similarity measure

In order to be able to express the power of the similarity measure used in the profiling step we need to design some evaluation criteria. Basic requirements are: to be able to compare our similarity measure with the "ideal" model that would always provide correct answers (in terms of identification based on behaviour characteristics) and the ability to compare results based on various input conditions (e.g. selected communication port, number of distinct source and destination addresses, ...). For example, we would like to see the influence of vector size on the similarity measurement in order to discuss "how much" information is still enough for the model to work "reasonably". By "reasonably" we mean the situation when (based on the similarity measurement) we are still able to pinpoint source IP address among others. Second big question is the use of IDF values. Does the idea that IDF values make the model more accurate hold for majority of cases? In this section we will describe the approach we use to evaluate a success ratio for a specific input setting (mainly application of optimisations with the input data).

The evaluation method should provide the ability to express the "amount of error" the model did in the similarity measurement phase. This would allow for comparison with the "ideal" model that would always produce correct answers (in terms of the similarity index).

First thing to do when starting with the evaluation is to normalise the calculated similarities for each source IP address so that the sum of all similar IP addresses equals 1. For a given source IP address, the sum of all similarity indexes is computed and each similarity index is then divided by this sum. In table 1 and source IP address A the list of similar IP addresses would change to: 0.316015(A,1); 0.316015 (B,1); 0.316015 (O,1); 0.051953(E,1). Second step is to create a vector that describes the "ideal" correct decision that the model should have done. At this point it is necessary to mention that for now, we assume that there is always only one correct answer – for a given source IP address from the set of the training data there is (if exists) only one source IP address from the set of testing data that would match the original source. All other sources from the latter set should be marked 0 by the similarity measurement process. We believe that with the data we have at the moment, this assumption and its application is the only way the model can be reasonably evaluated. Use of port 22 (SSH service) allows us to compare results with the correct information. This is done due to the fact that having this rule applied, the input data is not very noisy and behaviour based on the use of this service is more consistent.

Based on the prediction that we know the correct answer, we can create so called "correction vector" (i.e. the vector representing the ideal similarity) for each source IP address. The vector contains value 1 for the correct answer and 0 for all other, non-similar IP addresses.

To measure the amount of error for a given instance, we calculate the following *error_rates*:

$$error\_rate\ (good\ decision) = |1 - sim\_index|, \tag{3.6}$$

$$error\_rate\ (bad\ decision) = |0 - sim\_index|. \tag{3.7}$$

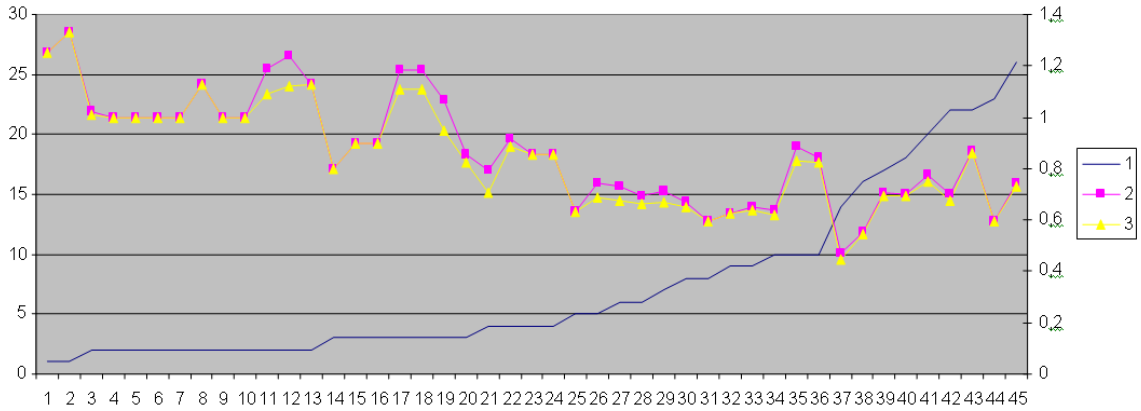


Figure 3.1: A graph correlating the behaviour vector dimensions (line 1) with amount of error (line 2 is without the influence of Inverse Document Frequency while line 3 is influenced by these values). Input restriction is minimal number of communication for both tables.

So for example for the source IP address A the *error_rates* would be 0.683985(A,1); 0.316015 (B,1); 0.316015 (O,1); 0.051953(E,1) – IP address A would have been the correct answer so the correction vector is (1, 0, 0, 0). Having this done we calculate a simple sum of all *error_rate* for a given source IP address and this number then represents the actual "amount of error". Boundaries within which this value can fall range from 0 – indicating the ideal correct answer to 2 – indicating the situation where the correct answer was marked as incorrect and some other source IP address was marked as the correct one. As we mentioned before, the correct answer is marked by 1 in the correction vector but there is a space to apply some threshold values and indicate the correct answer for example by 0.9.

Values representing the average amount of error are then used to evaluate correlations with different input settings and conditions.

Regarding the input conditions we use in this section to evaluate the similarity measure, first approach is to restrict the amount of minimal required communications from any source. We set five levels quintuple from 5 to 25. The same restriction is applied on both the training and the testing sets. Problem that this restriction causes is a speed with which the dimensions of the behavioural vectors decrease. The other approach is to apply that restriction on the first set only, then get the list of all destination IP addresses involved and use it as a restriction to build the testing set. This approach slows down the significant dimensions decrease, but the question of its influences on the accuracy of the model remains open.

In order to answer these questions, we performed similarity computations with several tables (month aggregates). Initial conditions (besides those mentioned in the previous paragraph) were the port 22 (SSH) and only one faculty. In general, for the evaluation purposes, we had to select those network segments where IP addresses do not change. Otherwise, we would not be able to see whether the model decision was correct or not. Therefore we do not consider wireless segments of the network as well as wired segments where laptops can be connected.

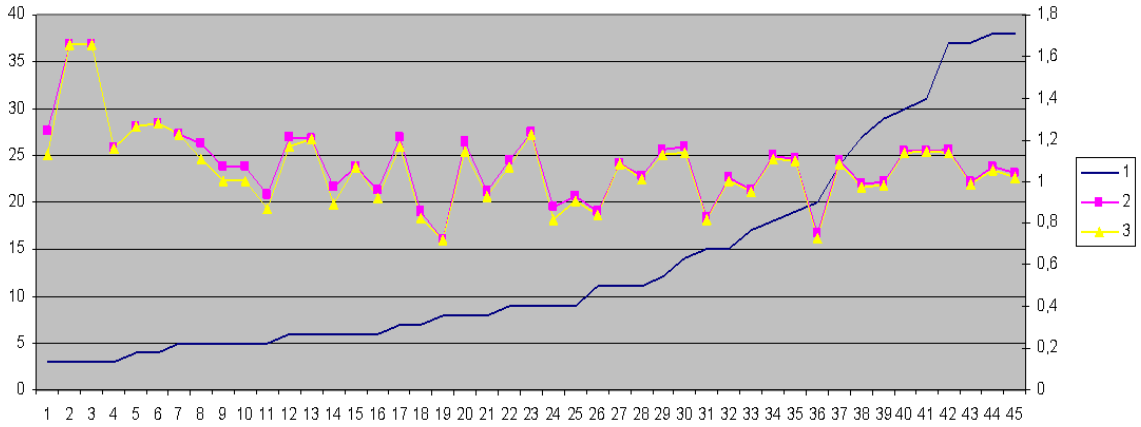


Figure 3.2: A graph correlating the behaviour vector dimensions (line 1) with amount of error (line 2 is without the influence of Inverse Document Frequency while line 3 is influenced by these values). Input restriction is minimal number of communication for first table and a list of involved destination IP addresses for the second table.

We did the similarity measurements for all those tables and plot important values to find the correlation between the vector dimensions (which actually represents the amount of information the model worked with) and the accuracy of the computation itself (average *error_rate* values).

Figure 3.1 shows the progress of the "amount of error" criterion correlated with the vector dimensions. It is quite clear that the use of IDF values makes certain improvement over values based on original behavioural vectors. Initial conditions were the minimal required number of connections for both the training and the testing sets. The same correlation but with different initial conditions (minimal required number of connections from the first table and a list of all involved destinations and respective source IP addresses from the second table) is shown on the Figure 3.2. If we compare both figures (Figures 3.1 and 3.2) we will see that in the former case the amount of error is in general slightly lower (Figure 3.3). Further experiments confirmed this trend and therefore it contradicts our original assumptions that by using the list of involved destinations we will put more data (in terms of the vector dimensions) into the similarity measurement and therefore the model would be more accurate. Having these questions answered we can try applying some threshold values to manipulate outputs of the similarity calculations. The idea is that if the answer is correct than the index of similarity is definitely over 0.5. Therefore we can try different thresholds to remove low indexes of similarity and observe how different thresholds influence the overall "amount of error" distribution. Being aware of the significance of the IDF values, from now on (if not stated otherwise) all results will be based on this approach.

Figure 3.4 summarises the influence of different threshold values applied on the similarity index. We start with 0.1 (which means we remove all similarity indexes below this value) and continue by 0.1 steps. With increasing threshold the accuracy of the model increased as well.

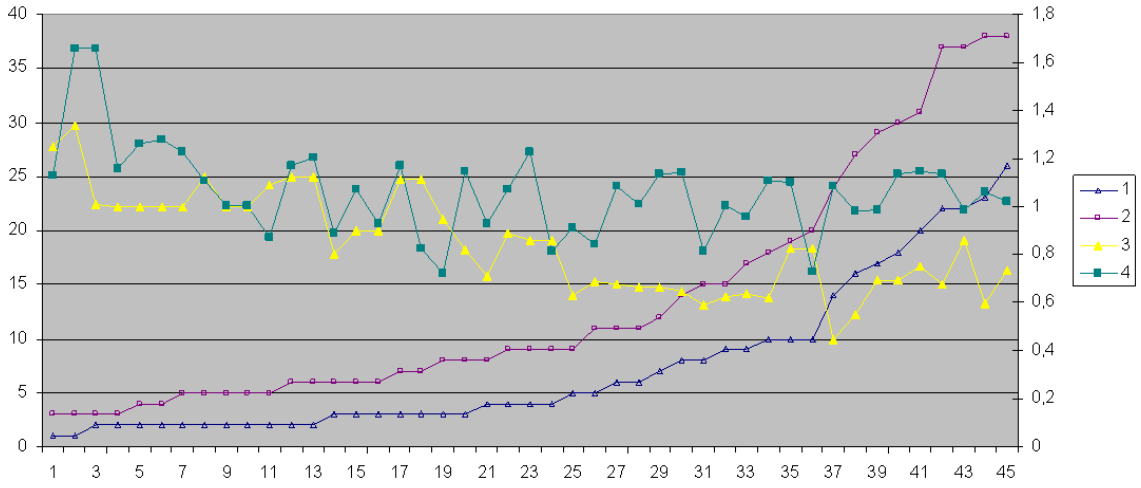


Figure 3.3: A graph comparing situation from the Figure 3.1 and the Figure 3.2. Line 1 – behaviour vector dimensions (both tables with the same input restriction). Line 3 – the corresponding trend of the "amount of error". Line 2 – behaviour vector dimensions (first table – the minimal number of communication; second table – the list of destination IP addresses). Line 4 – the corresponding trend of the "amount of error".

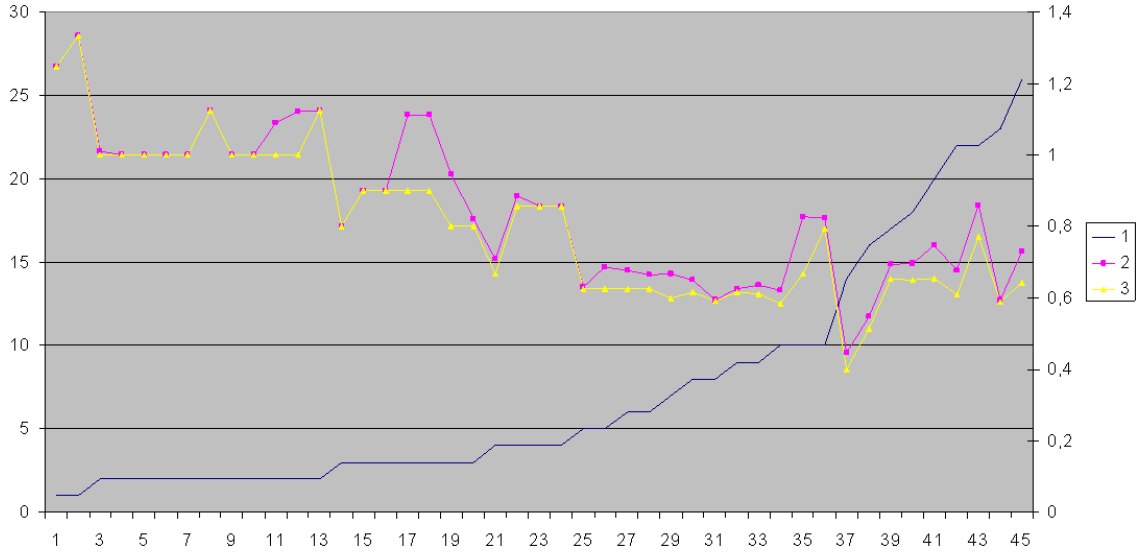


Figure 3.4: A graph comparing trends of the "amount of error" influenced by a threshold applied on the similarity index. Line 2 – no threshold, line 3 – threshold with the maximal influence (0.4).

To make the figure well-arranged, we show the accuracy without any threshold and with the threshold with the maximal impact in terms of accuracy. We found that the optimal value to be applied is around 0.4. Further increase could have influenced the correct answers which we definitely want to avoid so for this scenario, possibly the maximal accuracy is achieved by using the IDF values and setting the similarity index threshold to 0.4.

3.1.4 Conclusion

This section looked into the issue of profiling based on user past activity in a network. Our experiments with profiling have been based on data that is collected regularly in a large university network with thousands of connected users. This source can be considered ideal for our purposes, because it allows for building long-term behavioural characteristics that we assume to be the only identifying information of a user. The goal of the work presented was to answer the question about the chance that a user can be pinpointed among others based on her previous behaviour only.

In the first parts of the section we described methods for data pre-processing in order to analyze specific pieces only. The main problem is that the data in its entirety is too large to work with and therefore some kind of pre-processing is of a great importance. We described methods we use to decrease both the set of source and the set of destination IP addresses. We use Inverse Document Frequency to calculate the actual relevance which is highly dependent on the actual form of either set of IP addresses. Frequency histogram clustering is used to identify different sets of source IP addresses activity and is based on the assumption that if there is a similarity in the behaviour, then there should also be some similarity in frequencies of communication. However, in the evaluation, we started with much simpler sets of initial restrictions.

Once we get our input data prepared, we can continue with the profiling (or similarity measure). The methodology we use is based on the cosine similarity measure with addition of the actual relevancy of all destination IP addresses involved. Cosine similarity measures the actual angle between two vectors of the same dimensions. In our case, the vector is the information about the past activity of one source IP address. Outcomes of the similarity measurement are lists of similar source IP addresses – the original behaviour is based on a “training dataset” and similar IP addresses are searched in the “testing set”. We discussed all relevant details about the measure we use and showed some basic results for a better understanding.

The final part of our section is dedicated to the evaluation of the similarity measure. This step is natural in order to asses the descriptive power of the measure. We designed our own evaluation approach, where we can observe the actual “amount of error” of a given instance and observe the model performance under different circumstances. Based on our experiments, we confirmed our initial assumption that the use of IDF makes the model more accurate and we can do even better by setting reasonable thresholds.

In the environment we described in this chapter and based on our experiments we done so far, we can say, that if user behaves more or less consistently (during at least two months), we can distinguish his profile with considerably high accuracy.

For the following work, we plan to continue evaluating the model under different initial conditions and thus we will feed it with fairly high amounts of data. Then it would also be interesting to observe the model being fed with completely "unknown" data. During our current evaluations we know the correct answer that the model should have returned. This fact was used to calibrate the model and to observe its performance.

3.2 De-anonymisation of the Netflix Prize dataset

Micro-data can be used to profile or identify persons, as shown in the previous section. The idea is that in sparse databases any two records would not be similar. That is, the sparsity of the database expresses the individuality of persons, if each record in the database describes attributes, properties, behavior, or preferences of a person (and one and the same person is described by only one record). The individuality of the records does not even disappear when the records are sanitised, anonymised and perturbed.

In the previous section, network connection behavior of users has been utilised for profiling and determining individuals. A similar approach has been exercised by Narayanan and Shmatikov[NS08]. Instead of network traffic, Narayanan et al. analyzed the Netflix Prize dataset, a set of records released by Netflix, a large online DVD rental service, for improving the collaborative filtering recommender algorithms. The records contain the movie ratings (with corresponding timestamps) of a limited set of Netflix subscribers. The database sparsity arise, since there are large amounts of movies available, certainly many more than ordinary subscribers could watch or rate. Thus, the released data set contains the individual movie preferences of all included Netflix subscribers. Netflix anonymised the records by eliminating apparent references from records to persons. However, it was not possible to obfuscate the subscriber's preferences, since the they are basically necessary for the research on the collaborative filtering recommender algorithm, the actual purpose of the data release.

Instead of deducing the identity of Netflix subscribers from the released records, Narayanan et al. turned the de-anonymisation approach upside-down. The idea is that if you know a few details about a person's movie preferences, you may learn a lot more by identifying the corresponding Netflix subscriber record in the released data. Indeed, this only works out, if the person was a Netflix subscriber and the record is included in the released data. However, if it works out for a significant number of Netflix subscribers, it is perfectly clear that this de-anonymisation was neither intended by Netflix, nor was it in the first place considered by the subscribers. Thus, the de-anonymisation highlights a thread for the privacy of people which can presumably be generalised to the release of micro-data of any kind.

In addition to the knowledge which an adversary may learn from the plain Netflix Prize dataset, an adversary may also use the re-identified Netflix record as an additional identifier for the target person. With the identifier, the adversary would be able to learn even more data from establishing links from other datasets to the identifier, as Narayanan et al. point out.

The privacy threat which arises from sparse data is not easy to assess. Standard privacy metrics, such as k-anonymity[Swe02, KM07b], fail due to the high dimensionality of sparse data and, in particular, since the individuality is not only represented in a small number of dimensions, but rather in the interplay of them all: it is basically not possible to determine a *quasi-identifier* which does not include all dimensions. These problems are discussed in detail in [Agg05, MKGV07].

In order to re-identify Netflix subscribers, a rather technical link is required between the known details about the target person (auxiliary data) and the corresponding Netflix subscriber record. Narayanan et al. provide this link by means of the composition of three functions, the *record selection function* which works upon the records filtered by the *matching criterion* that filters records by means of a *scoring function*. The record selection function outputs the record with the best score among those records which comply with the matching criterion. The matching criterion ensures that only records with a score upon an appropriate threshold are considered for the record selection. This is to avoid false positives, that is records which are unlikely to be related to the target person, but are still the likeliest in the set of records. The scoring of each record is determined by the similarity of the record compared to the auxiliary data, that is the known details about the target person.

Supposed that all three functions were chosen appropriately, then it is very likely that the record returned by the record selection function is the Netflix subscriber record that belongs to the target person. In that case, the adversary learns from a limited set of a person's movie preferences, the auxiliary data, about all movie ratings that have been submitted to Netflix by the same person. If, however, there is no record returned from the record selection function, that is no record satisfied the matching criterion, then it is very likely that there is no record in the Netflix data which belongs to the target person.

The implementation of the record selection, the matching criterion, and the scoring function are described in detail in [NS08]. The scoring function is quite similar to the degree of similarity described in Section 3.1.2. It is also based on the cosine similarity and measures known from data mining such as the *support* of a record. In the following, we will not elaborate on the mathematics behind these functions. Instead, we focus on the conclusions of Narayanan et al. with regard to the Netflix Prize dataset.

Narayanan et al. investigated the question, "how much does the adversary need to know about a Netflix subscriber in order to identify her record if it is present in the dataset, and thus learn her complete movie viewing history?"[NS08] The findings of Narayanan et al. are rather alarming with regard to the privacy of Netflix subscribers that are included in the dataset. All records have been analysed. In order to obtain appropriate auxiliary data, a variant of each record has been used instead, that is the record itself, but limited to a smaller number of attributes and even with wrong attributes included. Intuitively, movie ratings for movies that are rarely rated contribute much more to the individuality of the record than frequently rated movies. But even without that distinction, 99% of the records can be re-identified, if eight ratings are known from the auxiliary data (two of them may even be wrong) and the date of the ratings is just to be known up to a 14-day error. With two ratings including the dates up to a 3-day error, still 68% of the records can be re-identified. The former number decreases

to 84%, if all movies in the auxiliary data are chosen from the top 500, that is from the 500 most rated movies. However, in fact each record in the data set includes at least one rating for a movie which is not among the top 100 and 97% of the records still include a rating for a movie which is not among the top 1000. In addition, 93% of the records include at least ten ratings for movies which are not among the top 100 and still 70% include at least 10 ratings for movies among the top 1000, cf. [NS08]. Thus, the distinction between frequently and rarely rated movies plays merely a role in theory, but not in practice with regard to the Netflix Prize dataset. Additionally, auxiliary data does not necessarily need to be a record. Narayanan et al. show that even the knowledge about the number of ratings may raise the probability of re-identification to the double. Surprisingly, this holds even if there is an error of up to 50% in that number.

4 Data Retention

The directive 2006/24/EC (data retention directive) "on the retention of data generated or processed in connection with the provision of publicly available electronic communication services or of public communication networks", passed by the European parliament on March 15th, 2006, sets the legal framework of data retention for the European Union member states. According to the directive, the member states have to "bring into force the laws, regulations and administrative provisions necessary to comply with this directive by no later than 15 September 2007" [2006/24/EC]. The goal of the directive is to strengthen the success of law enforcement in the area of Internet-related crime and, that is whenever electronic communication is involved. The motivation for the directive was that data about past communication relations is already unavailable when it comes to a trial after weeks or months where analysing the communication relations would certainly be of help. The relation data could provide information about the person who accessed a specific website or who called a specific telephone, for instance.

In the following sections, we discuss and measure the influence of data retention on anonymity services and particularly on the identifiability of their users.

4.1 Surveillance of Telecommunication in Germany

Surveillance of telecommunication in Germany is basically carried out in two ways:

1. Complete surveillance of telecommunications (e.g. by intercepting phone calls), based mainly on Art. 100a and b of the German Code of Criminal Procedure[1]
2. Evaluation of traffic data of telecommunications (e.g. connected numbers or IP-addresses, IMEIs, time and length of the communication), based on Art. 100g and 100h of the German Code of Criminal Procedure

By order of the German Federal Ministry of Justice the Max-Plank-Institute for foreign and international criminal legislation carried out two studies on both aspects [ADK03, AGK08]. These studies used a number of methods, especially:

- Evaluation of statistical data
- Evaluation of files concerning real cases from four federal states in Germany and
- Structured interviews with policemen, state attorneys and judges.

[1] German title: Strafprozessordnung (StPo), see `http://bundesrecht.juris.de/stpo/`

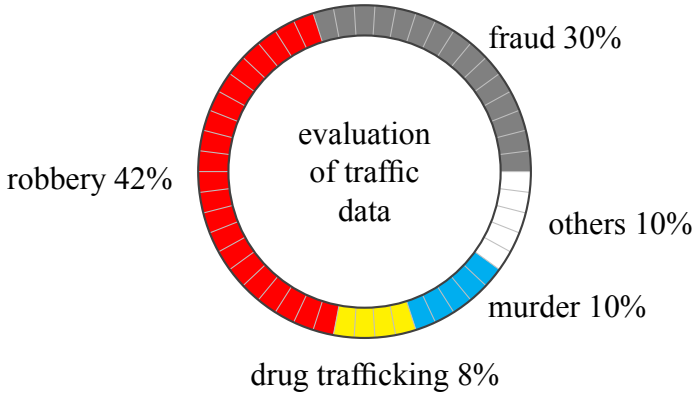


Figure 4.1: Reasons for evaluating telecommunication traffic data in criminal investigations in Germany since 2000.

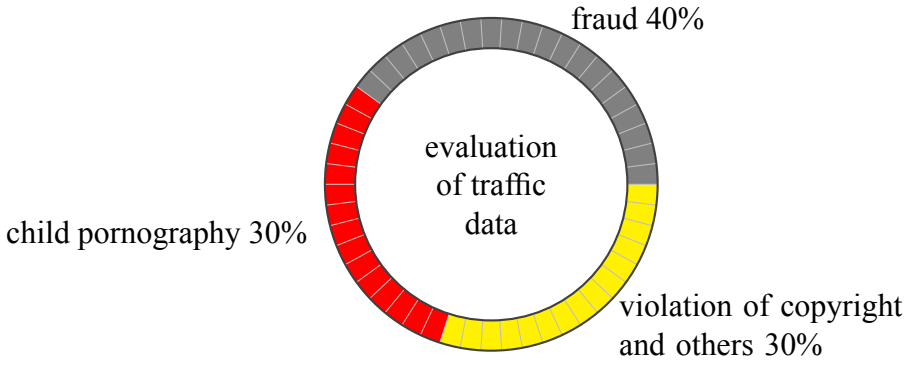


Figure 4.2: Reasons for evaluating telecommunication traffic data with respect to internet related cases in Germany since 2000.

Traffic data since 2000 has been used in criminal investigations increasingly. While in 2000 only 4000 requests of all types of surveillance by the police and state attorneys were reported, in 2005 it were already more than 40000. Mainly these requests concerned past traffic data. This included mainly incoming calls (60%), incoming and outgoing calls (22.5%) and searches in radio cells[2] (18%). Fraud using telecommunication equipment (30%), robbery and murder (ca. 10% each) and trafficking of drugs (8%) were mostly the reasons for the evaluation of traffic data of telephone communication, cf. Figure 4.1. With respect to internet communication fraud (more than 40%), children related pornography (ca. 30%) and violation of copyright were the most important reasons for surveillance, cf. Figure 4.2.

The time span covered by the search was as three month on average. Mostly the search was requested by the police (90% of investigated cases) and in 55% of these cases substantially

[2]Searches in radio cells aim at finding out who was in the cell (and thus a certain area) in a certain time span. As mobile phones logged in a cell are not logged in Germany, the assignment of a mobile phone to a certain cell is logged only in case the phone is used for incoming and outgoing calls. For that reason searches in cells are always based on connection data of calls.

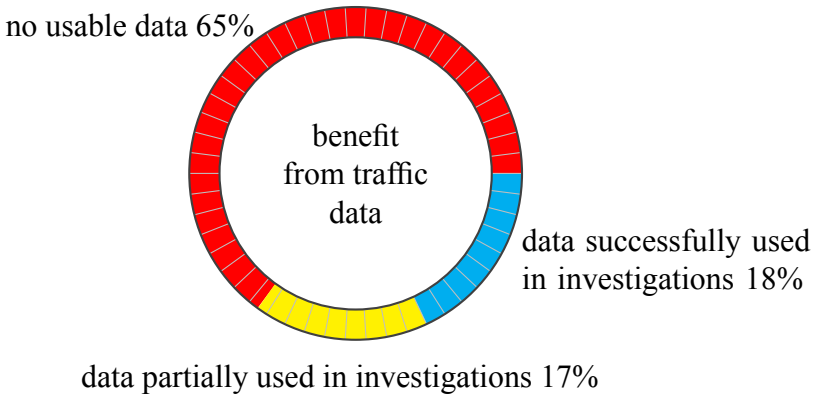


Figure 4.3: Benefit from telecommunication traffic data in criminal investigations in Germany since 2000.

justified from the point of view of the authors of the study. In 58% of the investigated cases the files were closed. 21% of the cases were brought to court; in 2% of the cases a penalty order was enacted. The remaining cases were not finally dealt with during the study. The court cases resulted in 85% in a conviction of which 16% included only serious crime resulting in imprisonment of 5 years or longer.

The German Code of Criminal Procedure defines surveillance of telecommunication including analysis of traffic data as a privacy restricting method to be used proportionally in cases where other methods were not successful or can not likely be used successfully. Though policy makers, policemen and state attorneys consider evaluation of traffic data less privacy invasive compared to interception of telecommunication, the German Code of Criminal Investigation requires in any case an assessment whether alternate, less privacy restricting investigation methods are available. The authors came to the conclusion that the analysis of traffic data was one of the first measures used by the criminal investigators. The reason was that in these cases the crime was committed using communication devices and infrastructure (mainly cases of fraud) and no other investigative methods were available. In other cases the evaluation of traffic data gave valuable hints for further investigations, especially in cases of murder. This also explains the success rates of the analysis of traffic data. 43% of the evaluations generated usable data, in 18% of the files analysed these data was successfully used in the investigation. 17% of the evaluations in the investigated cases were partially successful, 65% were not successful, cf. Figure 4.3. In court these data were mainly not mentioned, but the evaluation of traffic data lead in many cases to further evidence that was used in court instead.
In 37 investigated files data already was deleted (2% of the investigated files), in 17 cases partially anonymised. In the investigated files the use of cryptographic techniques (obviously anonymisation services are meant in this context) played no role. The interviews in addition supported this conclusion; no case was mentioned in which the used anonymisation services had a significant impact on the overall investigation.
Based on the evaluation of criminal investigation files it was concluded by the authors that especially the search in radio cells was invasive as each search concerned as an average

111 phone numbers in the cell (identified by outgoing calls) and 183 phone numbers for incoming calls.

The authors also made a number of recommendations. They suggest clarification of the legal norms by the legislator and standardisations in the proceedings for example based on a Code of Practice. In addition they point out that for the legally required effective decision by a judge sufficient personnel resources are required. The prior study from 2003 already came to the conclusion that the judges had not enough resources to check the requests for interceptions thoroughly. In the meantime no significant change in the resource situation has become known.

4.2 Data Retention and Anonymity Services

In this section we study the influence of the directive on the privacy of Internet users with a focus on the users of anonymity services. In our study, we use the legal framework provided by the German implementation of the data retention directive[3] and AN.ON as anonymity service. AN.ON has been developed at our site and is now successfully running on the Internet for years. The user base grew since the start of the service and is now reasonably large[4].

Germany has reacted to the directive and adapted several laws [Bun07a]. With respect to anonymity services on the Internet, the changes of the Telecommunications Act are most significant [Bun07b]. This act defines in detail what kind of data has to be stored for various types of communications providers, classical providers like fixed-line or mobile phone providers as well as modern ones like Internet service providers (ISPs). According to the act, the retention period is six months. The act anticipates services like anonymity services which are in the first place contradictory to the law enforcement goals. In order to prevent any information gap, the Telecommunications Act declared in §113a 'Retention of Data':

> '(6) Those, who provide telecommunication services and thereby alter data which have to be stored according to this law, have to store the original data and the new data as well as the time of the alteration.'[5]

Before examining the influence of this obligation in Section 4.2.4 and 4.2.5, we give a brief technical introduction of anonymity services with a focus on our AN.ON system in Section 4.2.1. We will concentrate on those aspects which are necessary to understand the following study on the influence of the data retention on the anonymity of AN.ON users. In

[3] So far, Germany is the only country that put the data retention directive into national legislation.

[4] The number of AN.ON users can be estimated by the traffic, but not finally determined, since we record no data which allows to discern different users.

[5] Note that the quotations of the Telecommunications Act is an unofficial translation of the official law text in German. The authors are not aware of any official translation of the current version of the Telecommunications Act. The former version (of 22 June 2004) is available in English (online at: `http://www.bmwi.de/BMWi/Redaktion/PDF/Gesetz/telekommunkationsgesetz-en`).

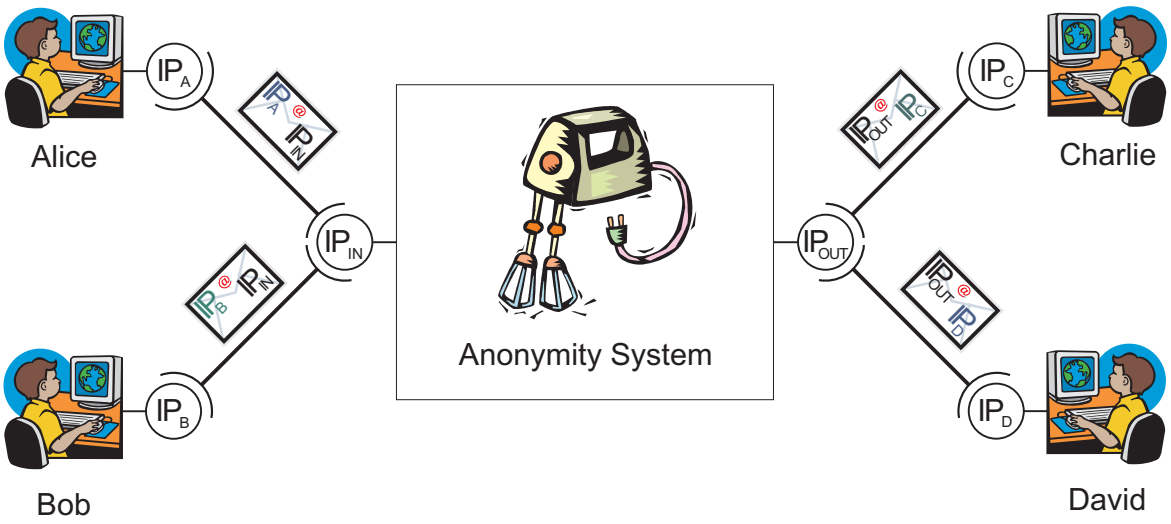


Figure 4.4: Anonymity service modelled as "black box" which replaces IP addresses of forwarded messages.

addition, we describe in Section 4.2.2 and 4.2.3 the attacks which we expect to be mounted on the retained data in order to compromise the anonymity of AN.ON users.

4.2.1 Anonymity services in a nutshell

For the matter of or study, we can understand an anonymity service as a "black box" which acts as a proxy, cf. Figure 4.4. Users redirect their network traffic through the proxy in order to achieve anonymity. Browsing the web without revealing the identity is a common application of anonymity services, for instance. In this context, we understand anonymity as the obfuscation of all relations that let an outsider, the adversary, learn about the links between incoming and outgoing proxy traffic. Consequently, the adversary would not be able to determine the persons who are exchanging messages through the proxy, if anonymity is preserved. This should even hold, if the adversary eavesdrop the data on all communication lines of the proxy. The anonymity of an anonymity service can be achieved by a combination of cryptography and data handling methods, such as padding, reordering, delaying etc.

In terms of sentence (6) of §113a of the Telecommunications Act the proxy, that is the anonymity service, replaces the IP addresses of senders and receivers with the proxy IP address in order to relay messages, cf. Figure 4.4.

An urging question is which data has to be logged by anonymity services such as AN.ON in order to comply with the data retention law. In §113a, the Telecommunications Act distinguishes several types of service and defines for each service the sort of data to be stored. The best match for AN.ON is 'Internet Service Provider' (ISP). According to the Telecommunications Act, an ISP has to log the IP address of a user, a unique identifier of the used connection, and the period of time in which this assignment was valid. In combination with

Sentence (6), this means that the anonymity service has to log the replacement of IP addresses only, but nothing more, particularly no 'identifiers' of higher layers, such as TCP-port numbers etc. Besides, consulted lawyers argue that only the replacement of source IP addresses (but not destination IP addresses) are allowed to be retained.[6] They justify their assessment with Sentence (8) of §113a: '…data about retrieved Internet pages must not be retained.' The lawyers also conclude that logging is allowed only for IP packet flows in upstream direction, that is only for packets from the user to the service, a web server for instance, but not for downstream packets. In fact, the effective interpretation of the law remains uncertain until the German Federal Supreme Court decides finally.[7] For this study, we assume that our interpretation is correct. Thus, we can derive that anonymity services have to log the replacement of the original source IP address whenever an IP packet is forwarded from a user to a server. In other words, the anonymity service has to log the time and source IP address of *every* IP packet it receives from a user.

4.2.2 Cross-section attack

Looking from a law enforcement point of view, the reply to the typical law enforcement question, "Which person was controlling IP address $\mathrm{IP}_{\mathrm{OUT}}$ at time t?" (Q_1), would include all logged source IP addresses for time t. Let $\mathcal{S}(t)$ denote the set of logged IP addresses. The size of $\mathcal{S}(t)$ depends on two parameters: (a) on the extent of usage of the (anonymity) service and (b) on the accuracy of the timestamp t. Note that the timestamp is not determined by law.

We have quantified the activity of users[8] of our AN.ON system in order to get a better idea of $\mathcal{S}(t)$ and its size. In order to keep the workload low, we decided to log the start and end time of *anonymous channels* only. The alternative would have been to log all incoming IP packets, but that would be rather expensive. In AN.ON, anonymous channels are the basic end-to-end communication vehicle, as a TCP/IP connection is on the network layer.

We found that nearly half of all channels lasted not more than one second, so we assume that analysing the channel activities leads to a good approximation of the actual size of $\mathcal{S}(t)$. Figure 4.5 shows the results of the quantification at the 'Dresden–Dresden' cascade of our AN.ON system.[9] The red dots depict the total number of users logged in, regardless if they were active or idle. The black dots show the number of users with at least one open channel.

[6] Even if this assumption does not hold, we see no benefit from retaining the destination IP addresses from a technical prospective. This is due to the fact that in reality anonymity services consist of a *sequence* of anonymity proxies.

[7] Other interpretations of the law can be found in the literature, e.g. in [PK08] the authors assume that: "the German legislation requires operators of anonymisers to link all incoming and outgoing messages and store this relation."

[8] When we speak about 'users' (e.g. number of users, activity of users etc.), we in fact talk about processes which speak the protocol of the AN.ON system and are connected to them. It is indeed not possible to say how many different human beings employ a single process.

[9] Our AN.ON system is based on MIX cascades. A cascade is a fixed chain of anonymity service servers (called MIXes). Users may freely choose the cascade they want to use.

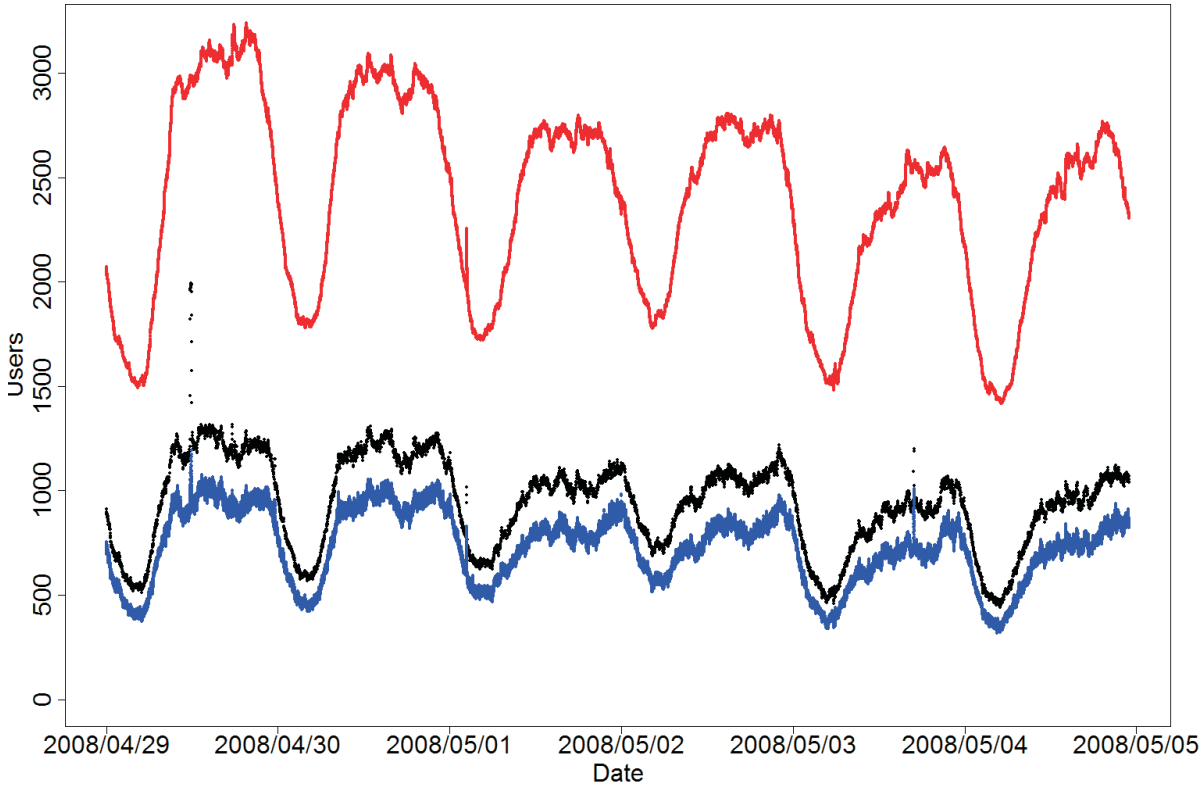


Figure 4.5: Graphs showing the total number of users logged in (red), the number of users which have an open channel within a given minute (black) and the number of users with an open channel within a given second (blue)

For both aggregations, a time resolution of *one minute* was used. For comparison, the blue dots depict a setting which is similar to the black dots, but with a time resolution of one second.

We see that the size of $\mathcal{S}(t)$, cf. Question (Q_1), has never been less than 400 between 29th of April 2008 and 5th of May 2008. That is, even when resolving a Q_1 request, a law enforcement agency would still have to investigate at least on 400 users to identify the person they are looking for. As we see in Figure 4.5, the accuracy of the time resolution (seconds vs. minutes, that is blue vs. black dots) is in practice less important than the overall usage rate of the anonymity service.

4.2.3 Intersection attack

The set of all online users at a single point in time might not be sufficient to narrow down the number of suspects to a reasonable small value, as we have seen in the previous section. Thus, a law enforcement agency could request the sets of online users for *several* points in time. With these sets, the agency would be able to mount an intersection attack [Ber99] which,

in theory, drastically narrows down the set size. Intersection attacks, however, require that the requested points in time are related to events that are linkable to one and the same target person. For the sake of simplicity, we assume that each event is related to exactly one point in time. Thus, a newly formulated question of a law enforcement agency would be "Which person was controlling IP address $\mathrm{IP}_{\mathrm{OUT}}$ at times t_1, t_2, and t_3?" (Q_2). If the law enforcement agency possesses a priori knowledge that one and the same target person is responsible for the events of interest observed at t_1, t_2, and t_3, then this person (or rather her identifier) is definitely in the intersection of $\mathcal{S}(t_1) \cap \mathcal{S}(t_2) \cap \mathcal{S}(t_3)$.

Note that events may basically occur on various layers, the application layer or the network layer, for instance. On the application layer, a law enforcement agency may observe that the same e-mail account was accessed several times. On the network layer, a law enforcement agency may run a honeypot and therefore obtain the exact timing of incoming IP packets which belong to one and the same TCP/IP connection.

4.2.4 Setup of our study on intersection attacks

In our study, we quantify the size of an anonymity set that remains after intersection attacks. The main problem with a study of intersection attacks on AN.ON user data is that (due to the very nature of anonymity services) there is no way to link the anonymised sessions of one and the same user. In order to get useful data for our study, additional identifiers need to be submitted by users to the AN.ON service.

We adapted the AN.ON client software such that users can decide whether they want to take part in the study. In the adapted software, a random number of 117 Bits is generated as identifier for those users who attend the study. The identifier is transmitted to AN.ON each time the user logs in to the Dresden–Dresden cascade. Thus, user sessions became linkable on a voluntary base within our study.

The identifiers have been recorded in the time between 21th of May and 20th of July, 2008. On 21th of May, the adapted client has been released and old clients conceive this as a necessary update. Thus, we expect that, in the following days, the vast majority of AN.ON users had installed the new client and were therefore asked whether they want to support the study or not. In total, we recorded 70591 replies, 38738 replies were positive, that is 54.88% agreed to support the study. The remaining 45.12% of the users continued to use AN.ON without any linkability of their sessions.

In addition to the symbol $\mathcal{S}(t)$ which we informally introduced in a previous section, we define the symbol $\mathcal{S}_{\cap}(T)$ as the AN.ON users which were logged in at all times $t_1, \ldots, t_n \in T$. The formal definitions are reflected in Equation (4.2) and (4.3).

Let I_u be the set of all user IDs, I_s be the set of all session IDs, $\mathcal{P}(I_s)$ the power set of all session IDs, and let $X : I_u \to \mathcal{P}(I_s)$ with

$$X(uid) = \big\{ sid \in I_s \;\big|\; sid \text{ related to } uid \big\}. \tag{4.1}$$

Additionally, let $t_{\text{in}}(sid)$ and $t_{\text{out}}(sid)$ be the time of login and logout, respectively, with regard to the session $sid \in I_s$. We define $\mathcal{S}(t)$ as the set of users which have been logged in at AN.ON within t and $t + t_{\text{res}}$ where t_{res} is the time resolution, that is a second or a minute in our study.

$$\mathcal{S}(t) = \left\{ uid \;\middle|\; uid \in I_u, sid \in X(uid), t_{\text{in}}(sid) < t + t_{\text{res}}, t_{\text{out}}(sid) \geq t \right\} \tag{4.2}$$

With $\mathcal{S}(t)$, we can define $\mathcal{S}_\cap(T)$, the anonymity set after an intersection attack with times $T = \{t_1, \ldots, t_n\}$. We suppose that all elements in T are pairwise different, that is for $T = \{t_1, \ldots, t_n\}$ holds $|T| = n$.

$$\mathcal{S}_\cap(T) = \bigcap_{t \in T} \mathcal{S}(t) \tag{4.3}$$

In the following, we focus on intersections between user sets of *two* points in time only. That is, we explore $\mathcal{S}_\cap(T)$ with the samples T where $|T| = 2$ and the elements of T are chosen by random[10]. This setting can be understood as the case that law enforcement agencies request the set of persons which have been logged in at t_1 and at t_2 as well, or in $T = \{t_1, t_2\}$, respectively.[11]

Unfortunately, a complete exploration of all possible intersections for the entire time of the study, that is (pairwise) intersections over more than 2.5 million points in time, would require more than 3 trillion intersections. This definitely exceeds our computing capacities. Thus, instead of exploring all intersections, we decided to choose sample sets randomly in the space of all points in time. For this analysis, 5 million of such samples have been explored.

4.2.5 Results of our study on intersection attacks

In Figure 4.6, we show three frequency density diagrams that show immediate results from our study with 5 million samples of two points in time t_1 and t_2. On the horizontal axis, we see the size of $\mathcal{S}(t)$ or $\mathcal{S}_\cap(T)$ for given t or T, respectively. On the vertical axis, we see the frequency densities of these set sizes with regard to our samples. The red line marks the frequency densities of set sizes of $\mathcal{S}(t_1)$. Accordingly, the green and the blue line mark the frequency densities with regard to $\mathcal{S}(t_2)$ and $\mathcal{S}_\cap(\{t_1, t_2\})$, respectively[12]. The parameters of the distributions are reported in Table 4.1. Similar results are shown in Figure 4.7, 4.8, and 4.9 and the corresponding tables with variations in (a) the time resolution and (b) the user data (login/logout vs. activity[13]). We see that the anonymity set size is greater with a coarser time resolution and, even more significantly, if the adversary has no access to the activity data of AN.ON users, but only to their login/logout behaviour.

[10] In the worst case, we are going to obtain an empty intersection set due to the random selection of sets. A law enforcement agency would indeed make use of auxiliary data such that the intersection would always contain at least one person.

[11] In practice, law enforcement agencies will rather request who used the IP address of the last MIX of a cascade

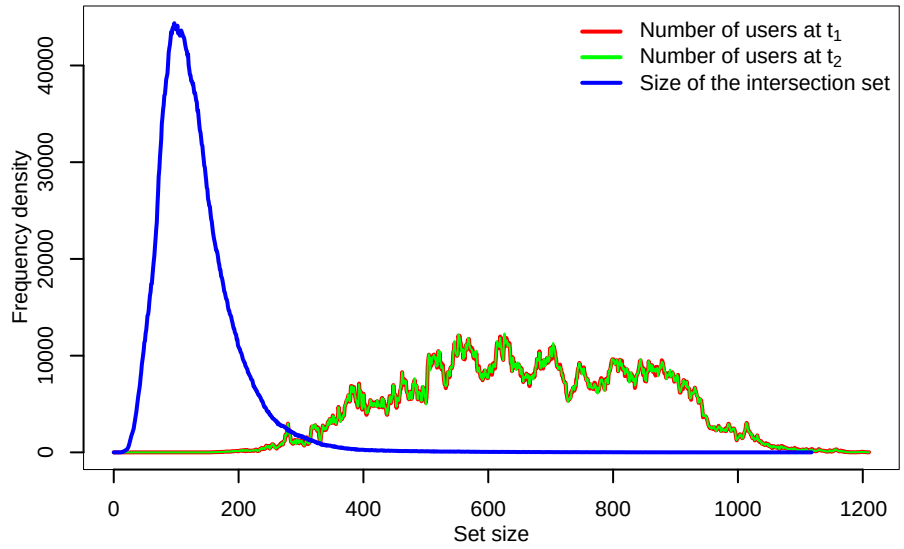


Figure 4.6: Login/Logout, time resolution of 1sec: Frequency density diagram of the anonymity set sizes $|\mathcal{S}(t_1)|$, $|\mathcal{S}(t_2)|$, and $|\mathcal{S}_\cap(\{t_1, t_2\})|$

	Min	1st Quartile	Avg	Median	3rd Quartile	Max
$\lvert\mathcal{S}(t_1)\rvert$	148	529	661	665.6	814	1210
$\lvert\mathcal{S}(t_2)\rvert$	147	529	660	665.5	814	1210
$\lvert\mathcal{S}_\cap(\{t_1, t_2\})\rvert$	9	91	120	133.1	159	1118

Table 4.1: Login/Logout, time resolution of 1sec: characteristics of the distributions of $|\mathcal{S}(t_1)|$, $|\mathcal{S}(t_2)|$, and $|\mathcal{S}_\cap(\{t_1, t_2\})|$

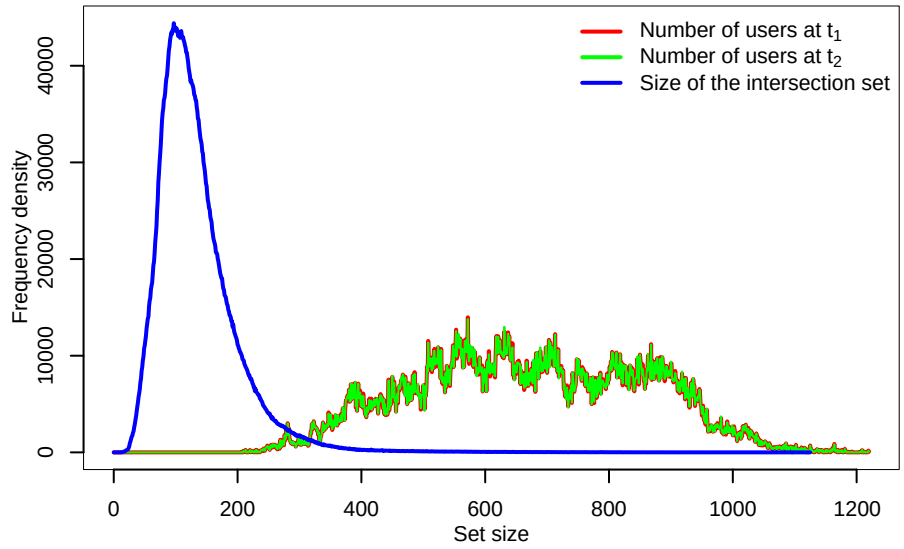


Figure 4.7: Login/Logout, time resolution of 1min: Frequency density diagram of the anonymity set sizes $|\mathcal{S}(t_1)|$, $|\mathcal{S}(t_2)|$, and $|\mathcal{S}_\cap(\{t_1, t_2\})|$

	Min	1st Quartile	Avg	Median	3rd Quartile	Max
$\lvert\mathcal{S}(t_1)\rvert$	207	534	667	672	821	1220
$\lvert\mathcal{S}(t_2)\rvert$	207	534	667	671.9	821	1220
$\lvert\mathcal{S}_\cap(\{t_1, t_2\})\rvert$	11	92	121	134.1	161	1125

Table 4.2: Login/Logout, time resolution of 1min: characteristics of the distributions of $|\mathcal{S}(t_1)|$, $|\mathcal{S}(t_2)|$, and $|\mathcal{S}_\cap(\{t_1, t_2\})|$

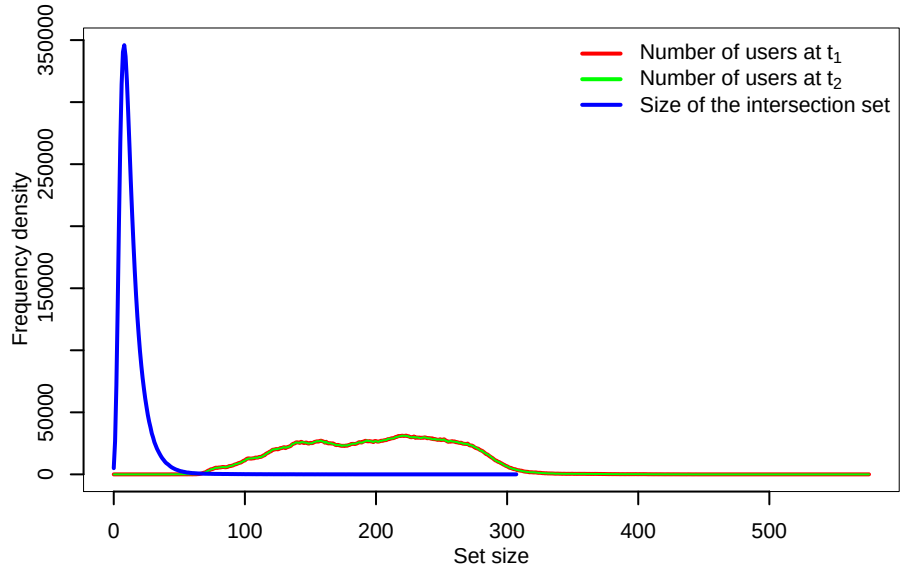


Figure 4.8: User activity, time resolution of 1sec: Frequency density diagram of the anonymity set sizes $|\mathcal{S}(t_1)|$, $|\mathcal{S}(t_2)|$, and $|\mathcal{S}_\cap(\{t_1, t_2\})|$

	Min	1st Quartile	Avg	Median	3rd Quartile	Max
$\lvert\mathcal{S}(t_1)\rvert$	23	153	202	199.2	245	576
$\lvert\mathcal{S}(t_2)\rvert$	23	153	202	199.2	245	576
$\lvert\mathcal{S}_\cap(\{t_1, t_2\})\rvert$	0	7	11	13.39	17	307

Table 4.3: User activity, time resolution of 1sec: characteristics of the distributions of $|\mathcal{S}(t_1)|$, $|\mathcal{S}(t_2)|$, and $|\mathcal{S}_\cap(\{t_1, t_2\})|$

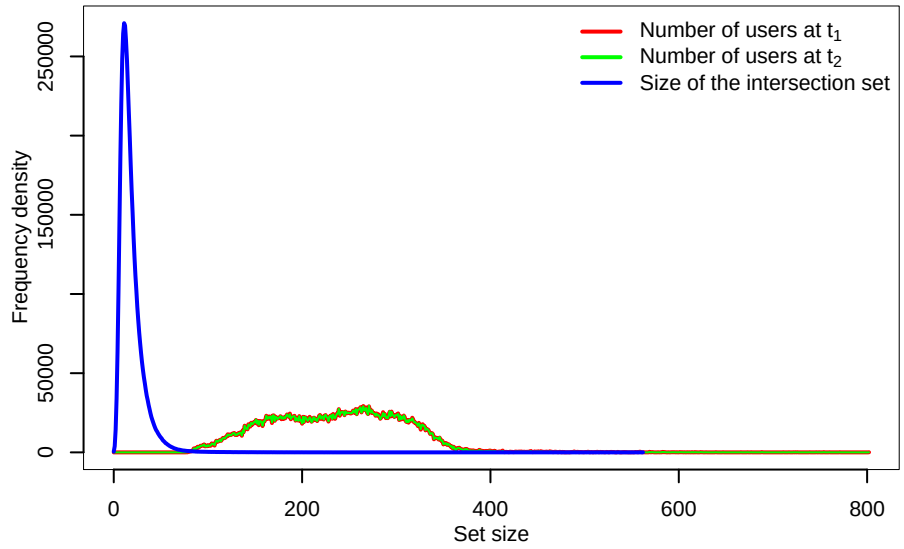


Figure 4.9: User activity, time resolution of 1min: Frequency density diagram of the anonymity set sizes $|\mathcal{S}(t_1)|$, $|\mathcal{S}(t_2)|$, and $|\mathcal{S}_\cap(\{t_1, t_2\})|$

	Min	1st Quartile	Avg	Median	3rd Quartile	Max
$\lvert\mathcal{S}(t_1)\rvert$	77	181	238	235.9	288	802
$\lvert\mathcal{S}(t_2)\rvert$	77	181	238	235.9	288	802
$\lvert\mathcal{S}_\cap(\{t_1, t_2\})\rvert$	0	10	15	18.42	23	562

Table 4.4: User activity, time resolution of 1min: characteristics of the distributions of $|\mathcal{S}(t_1)|$, $|\mathcal{S}(t_2)|$, and $|\mathcal{S}_\cap(\{t_1, t_2\})|$

	Model			
	1	2	3	4
Predictors				
Intercept	-2.05	-3.13	1.53	1.56
	(0.04)	(0.05)	(0.03)	(0.02)
minimum set size	1.09	1.00	1.00	0.98
$\ln\big(\min\big(\lvert\mathcal{S}(t_1)\rvert, \lvert\mathcal{S}(t_2)\rvert\big)\big)$	(0.01)	(0.01)	(0.00)	(0.00)
maximum set size		0.25		
$\ln\big(\max\big(\lvert\mathcal{S}(t_1)\rvert, \lvert\mathcal{S}(t_2)\rvert\big)\big\rvert$		(0.01)		
time interval			-0.22	-0.22
$\ln\lvert\delta\rvert$			(0.00)	(0.00)
periodicity indicator				0.24
$f_{\triangle}(\delta)$				(0.00)
Summary				
adjusted R^2	0.48	0.49	0.79	0.82

Table 4.5: Parameters of different linear regression models with dependent variable $\lg\big\lvert\mathcal{S}_{\cap}(\{t_1, t_2\})\big\rvert$. ($N$ = 5 Million data points). Std. errors in brackets and coefficient significance in 0.001 level.

We evaluated four regression models in order to analyse the causal relationship between the intersection size and various explanatory variables, cf. Figure 4.6 and Table 4.1. The parameters of these models are compiled in Table 4.5.

In Model 1, we try to explain the size of $\mathcal{S}_{\cap}(\{t_1, t_2\})$ by the minimum size of $\mathcal{S}(t_1)$ and $\mathcal{S}(t_2)$. The results are log transformed to reasonably normalise the residuals. Additionally, in Model 2, we add the maximum size of $\mathcal{S}(t_1)$ and $\mathcal{S}(t_2)$ as a second explanatory variable. We see that the gain of explained variance, cf. adjusted R^2 in Table 4.5, is small and the coefficient lower – albeit positive and statistically significant. This is what we expect, since the intersection set $\mathcal{S}_{\cap}(\{t_1, t_2\}) = \mathcal{S}(t_1) \cap \mathcal{S}(t_2)$ is at most as great as the smallest set of $\mathcal{S}(t_1)$ and $\mathcal{S}(t_2)$. As the set size fluctuates heavily over time, the size of the intersection is strongly related to the minimum size of $\mathcal{S}(t_1)$ and $\mathcal{S}(t_2)$.

In all following models, we drop the less-influentical maximum and use solely as to control for a varying number of users over time. In Model 3, the negative sign of the coefficient $\lg(\delta)$ indicates that there is an inverse relation between the time interval and the intersection size.

in T than requesting the login state of AN.ON users.

[12] Note that from a theoretical point of view the distributions of the set sizes $S(t_1)$ and $S(t_2)$ should be the same. We show them both in the following graphs and tables as a kind of plausibility check.

[13] With activity, we refer to channel activity as described in Section 4.2.2.

That means, smaller time intervals lead to greater intersection sets, since the smaller the time interval between t_1 and t_2, the higher is the likelihood that a user who is logged in at t_1 is still logged in at t_2. The considerable gain in R^2 of 31 percentage points reveals that time between events matters.

In Model 4, we explore the influence of user behaviour on the set size of $\mathcal{S}_{\cap}(\{t_1, t_2\})$. We expect that the user behaviour follows regular pattern, for instance a periodicity of 24h.[14] This is so because we expect that users pursue similar tasks at similar times of the day. Users who log in to AN.ON during the working hours may regularly use AN.ON in their profession, for instance journalists. Those users who use AN.ON for their leisure time activities may regularly log in after the working hours. In order to check the support of our expectation in the sample data, we estimate the coefficient of an indicator variable computed from a periodic triangular function $f_{\Delta}(\delta)$ which generates an indicator variable that yields a value between 0 and 1, where a value of 0 marks the smallest match with the 24h pattern and a value of 1 denotes the best match.

$$\delta = |t_1 - t_2| \tag{4.4}$$

$$f_{\Delta}(\delta) = \left| 1 - \frac{\delta \bmod (24 \cdot 60^2)}{12 \cdot 60^2} \right| \tag{4.5}$$

The positive coefficient indicates that the sample data in fact shows periodicity, although the conditional explanatory power of this sample linear function is rather small (3 percentage points).

4.2.6 Conclusions

The intention of our study is to assess the risk which arises from an attacker who is mounting intersection attacks on anonymity systems. We used data about user behaviour which has been accumulated at our anonymity system AN.ON that is hosted at TU Dresden site, among others. From the results, we see that hiding in an anonymity set works well as long as adversaries request single user sets that belong to a distinct point in time without relating them to each other. However, the results also show that an adversary who is combining the results of different requests, and therefore is requesting several anonymity sets in order to intersect them, has much more success in narrowing down individuals in the anonymity set. Compared to a single request, the intersection of only two requests reduces the size by far more than 50%. Though, this is hardly sufficient for law enforcement agencies that seek to reduce anonymity sets to single persons, the results can be further refined, presumably with similar success, by intersecting more anonymity sets that are known to contain the target person.

Our results show that there is a remarkable difference between the different ways to request data with regard to the sizes of the anonymity set. The anonymity sets are larger, if the set of those users is requested who were *logged in* in a distinct moment in time. The anonymity

[14] It was not possible to explore patterns on a weekly or longer basis, since our study period was too short

sets decrease in size, if only *active* users are requested. The anonymity sets are presumably smallest, if the requests do not address the application layer of the anonymity service, but the underlying network layer, however, our study was limited to the application layer.

Even though this discussion may lead to the conclusion that it is necessarily desirable for an adversary or a law enforcement agency to request user sets of active users only, this idea may be misguiding for anonymity systems such as AN.ON: Users may send dummy traffic as a countermeasure. The idea behind dummy traffic of users is to make themselves appear active, even though they are actually idling. This can be achieved by regularly sending data packages from the user to the service without any content of interest. It is indeed crucial that besides the user and the anonymity service, no one may be able to distinguish dummy traffic from ordinary traffic. Thus, if users send dummy traffic, a law enforcement agency which is able to obtain the set of all *active* users would not learn more than an agency which is limited such that it can only observe set of all users that are *logged in*.

Dummy traffic has been discussed with regard to several attack schemes [BL02, DP04a, DP04b]. In general, it has been found to be a rather weak countermeasure. However, due to the limitations of the "adversary" described by the data retention act, a continuous connection to the anonymity service together with dummy traffic seems a striking good solution. The economical aspects of dummy traffic have been mentioned in literature, but are of decreasing significance in a world with complete network coverage and flat rates.

5 Conclusions

In literature and in previous deliverables, perfect or at least reasonable anonymity measures have been proposed [KM07a, KM07b]. These measures make either strong assumptions about the power of the adversary or require an unrealistic anonymity service model, for instance the DC-Net. The outcome of this deliverable is twofold, (1) we see that in reality the assumptions of these rather theoretic anonymity measures do not fit properly and in common settings such as the Web 2.0, anonymity is quite hard to preserve, and (2) even though it might be possible to recover a lot of the information which is hidden by anonymising techniques, it is hard to determine what is necessary and practical in reality in order to recover distinct identities.

As to the first outcome, we have seen in Section 3.2 that personal data has been anonymised by the data holder with common techniques, but persons were easy to re-identify, even with just little auxiliary data and an astonishing high accuracy. In particular personal data about preferences of any kind and social behaviour is likely to be exploited, as also outlined in Chapter 2. Unfortunately, it is exactly this kind of data which users intentionally share and publish on the Internet in the Web 2.0. Though, similar results can be shown with data from the network layer, as shown in Section 3.1.

As to the second outcome, we have studied the effect of data retention on anonymisation services, cf. Section 4.2. In order to achieve realistic results, we used the conditions of data retention defined in the German Telecommunications Act. The act bases on directive 2006/24/EC of the European Community and defines what data has to be retained and what data must not. On the other end, we used AN.ON, an anonymisation service which is run at TU Dresden. That enabled us to avoid any steps in between, such as user models, and study real user data. We found that requests of a law enforcement agency for retained data would neither immediately lead to the re-identification of a person, nor to a acceptable decrease of the set of suspects. For re-identification, it would need two or more requests and an intersection attack on the retained data. However, intersection attacks require that all events that justify the requests of the law enforcement agency are linkable to the same target person. However, this is a rather tough condition, since the event itself may be an observed behaviour or property of the target person. But people change both over time. Thus, the more events are necessary to re-identify the person, the higher is the probability that the target person changed her habits and one of the requested data sets does not belong to her.

In sum, we see that it is quite hard for individuals to preserve their anonymity when acting on the Internet such as using Web 2.0 applications. On the other side, we see that is not trivial to setup laws that allow to re-identify target persons in a save manner.

References

[ADK03] H. Albrecht, C. Dorsch, and C. Krüpe. Rechtswirklichkeit und Effizienz der Überwachung der Telekommunikation nach den §§100a, 100b StPO und anderer verdeckter Ermittlungsmaßnahmen, 2003. http://www.bundesjustizministerium.com/files/-/136/Abschlussbericht.pdf.

[Agg05] Charu C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st VLDB Conference*, Trondheim, Norway, 2005.

[AGK08] H. Albrecht, A. Grafe, and M. Kilching. Rechtswirklichkeit der Auskunftserteilung über Telekommunikationsverbindungsdaten nach §§100g, 100h StPO, 2008. http://www.bmj.bund.de/files/-/3045/MPI-GA-2008-02-13%20Endfassung.pdf.

[BA99] Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[BE07] Danah Boyd and Nicole B. Ellison. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1-2), November 2007.

[Ber99] Oliver Berthold. Effiziente Realisierung von Dummy Traffic zur Gewährleistung von Unbeobachtbarkeit im Internet. Diplomarbeit, Technische Universität Dresden, Faculty of Computer Science, Institute for Theoretical Computer Science, 01062 Dresden, December 1999. in German.

[Ber03] Michael Berry. *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer, 1 edition, September 2003.

[BL02] Oliver Berthold and Heinrich Langos. Dummy traffic against long term intersection attacks. In Roger Dingledine and Paul Syverson, editors, *Proceedings of Privacy Enhancing Technologies workshop (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.

[Bun07a] Bundestag. Gesetz zur Neuregelung der Telekommunikationsüberwachung und anderer verdeckter Ermittlungsmaßnahmen sowie zur Umsetzung der Richtlinie 2006/24/EG vom 21. Dezember 2007. *Bundesgesetzblatt*, Jahrgang 2007(Teil I, Nr. 70):3198–3211, December 2007. ausgegeben zu Bonn.

[Bun07b] Bundestag. Telekommunikationsgesetz vom 22. Juni 2004, 2007. BGBl. I S. 1190, zuletzt geändert durch Artikel 2 des Gesetzes vom 21. Dezember 2007 (BGBl. I S. 3198).

[Cot] Lance Cottrell. Mixmaster & remailer attacks. Unpublished manuscript, "`http://www.obscura.com/~loki/remailer/remailer-essay.html`".

[DP04a] Claudia Díaz and Bart Preneel. Reasoning about the anonymity provided by pool mixes that generate dummy traffic. In *Proceedings of 6th Information Hiding Workshop (IH 2004)*, LNCS, Toronto, May 2004.

[DP04b] Claudia Díaz and Bart Preneel. Taxonomy of mixes and dummy traffic. In *Proceedings of I-NetSec04: 3rd Working Conference on Privacy and Anonymity in Networked and Distributed Systems*, Toulouse, France, August 2004.

[DS04] George Danezis and Andrei Serjantov. Statistical disclosure or intersection attacks on anonymity systems. In *Proceedings of 6th Information Hiding Workshop (IH 2004)*, LNCS, Toronto, May 2004.

[DTD07] Claudia Díaz, Carmela Troncoso, and George Danezis. Does additional information always reduce anonymity? In Ting Yu, editor, *Workshop on Privacy in the Electronic Society 2007*, pages 72–75. ACM, 2007.

[DTS08] Claudia Díaz, Carmela Troncoso, and Andrei Serjantov. On the impact of social network profiling on anonymity. In *8th Privacy Enhancing Technologies Symposium*. to appear in Springer, 2008.

[HB05] Mireille Hildebrandt and James Backhouse, editors. *D7.2: Descriptive analysis and inventory of profiling practices, Deliverable of FIDIS' Workpackage 7*. FIDIS `www.fidis.net`, 2005.

[JCAB08] David-Olivier Jaquet-Chiffelle, Bernhard Anrig, and Emmanuel Benoist, editors. *D13.8 Applicability of privacy models, Deliverable of FIDIS' Workpackage 13*. FIDIS `www.fidis.net`, 2008.

[KM07a] Marek Kumpošt and Vacláv (Vašek) Matyáš, editors. *D13.1 Identity and impact of privacy enhancing technologies, Deliverable of FIDIS' Workpackage 13*. FIDIS `www.fidis.net`, 2007.

[KM07b] Marek Kumpošt and Vacláv (Vašek) Matyáš, editors. *D13.6 Privacy Modelling and Identity, Deliverable of FIDIS' Workpackage 13*. FIDIS `www.fidis.net`, 2007.

[Kum07] Marek Kumpošt. Data preparation for user profiling from traffic log. In *Proceedings of The International Conference on Emerging Security Information, Systems, and Technologies (SECURWARE 2007)*, volume 0, pages 89–94, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[LB04] Jae-Woo Lee and Doo-Kwon Baik. A model for extracting keywords of document using term frequency and distribution. In Alexander F. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing, 5th International Conference, CICLing 2004*, volume 2945 of *Lecture Notes in Computer Science*, pages 437–440. Springer, 2004.

[MC04] V. Matyáš and D. Cvrček. On the role of contextual information for privacy attacks and classification. In *Privacy and Security Aspects of Data Mining Workshop*, Brighton, UK, November 2004. IEEE ICDM.

[MKGV07] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *ACM Transactions on Knowledge Discovery from Data (TKDD)*, volume 1. ACM, New York, NY, USA, March 2007.

[NS08] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proc. of 29th IEEE Symposium on Security and Privacy*, Oakland, CA, May 2008. To appear.

[PK08] Lexi Pimenidis and Eleni Kosta. The impact of the retention of traffic and location data on the internet user. *DuD Datenschutz und Datensicherheit*, 32(2):92–97, February 2008.

[Swe02] Latanya Sweeney. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[UMS03] Peter Palfrader Ulf Moller, Lance Cottrel and Len Sassaman. Mixmaster protocol - version 2. `http://www.abditum.com/mixmaster-spec.txt`, 2003.

[War63] Joe H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[WS98] Duncan Watts and Steven Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.