FIDIS

Future of Identity in the Information Society

Title:	"D13.8: Applicability of privacy models"
Author:	WP13
Editors:	David-Olivier Jaquet-Chiffelle Bernhard Anrig Emmanuel Benoist (VIP Bern University of Applied Sciences)
Reviewers:	Jozef Vyskoc (VaF, s.r.o. Slovak Republic) Hans J.C. Buitelaar (Tilburg U., Netherlands)
Identifier:	D13.8
Type:	[Deliverable]
Version:	1.0
Date:	Monday, 02 June 2008
Status:	[Final]
Class:	[Public]
File:	fidis-wp13-del13.8.pdf

Brief Summary

In the present deliverable, we focus on the applicability of privacy models and review as well as illustrate the applicability of models from Deliverable D13.6 using a real-world example. Besides, we show some shortcomings of the approaches presented in D13.6 and include the aspects of combination of information and of misinformation, i.e., information which (partly) cannot to some extend and for some reason be verified by an adversary, hence approaches which may potentially be of major influence in the computation of a measure of anonymity.

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

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	Germany
5. Institut Europeen D'Administration Des Affaires (INSEAD)	France
6. University of Reading	United Kingdom
7. Katholieke Universiteit Leuven	Belgium
8. Tilburg University	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	Czech Republic
14. VaF Bratislava	Slovakia
15. London School of Economics and Political Science	United Kingdom
16. Budapest University of Technology and Economics (ISTRI)	Hungary
17. IBM Research GmbH	Switzerland
18. Centre Technique de la Gendarmerie Nationale	France
19. Netherlands Forensic Institute	Netherlands
20. Virtual Identity and Privacy Research Center	Switzerland
21. Europäisches Microsoft Innovations Center GmbH	Germany
22. Institute of Communication and Computer Systems (ICCS)	Greece
23. AXSionics AG	Switzerland
24. SIRRIX AG Security Technologies	Germany

Versions

Version	*Date*	*Description (Editor)*
0.1	12.12.2007	• Initial structure, draft of chapter structure
0.2	22.03.2008	• First overview of chapters, initial draft for discussion • Discussion at the GM2008
0.3	07.04.2008	• First draft of chapters • Integration of Chapters 2, 4 and 5
0.4	11.04.2008	• Integration of Chapter 3
0.5	18.04.2008	• Editorial work, integration • Introduction and conclusion
0.6	25.04.2008	• Review by authors, updates of chapters • Integration, extension of introduction and conclusions
0.7	28.04.2008	• Additional editing work
0.8	30.04.2008	• First version sent to internal reviewers
0.9	29.05.2008	• Revision after internal review • Final check by authors
1.0	02.06.2008	• Final version

Foreword

FIDIS partners from various disciplines have contributed as authors to this document. Researchers from the following institutions were the main contributors for the chapters of this document:

- COSIC Katholieke Universiteit Leuven, Belgium (7)
- Masarykova univerzita, Brno, Czech Republic (13)
- Technische Universität Dresden, Germany (11)
- VIP Berne University of Applied Sciences (20)

Other contributions, and namely the feedback of internal reviewers, can also be found in this deliverable.

Management Summary

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

This deliverable must be seen in the light of a series of three deliverables within Workpackage 13, namely D13.6 *Privacy Modeling and Identity*, this one D13.8, and the forthcoming D13.9 *Estimating quality of identity*, which develop the subject of privacy and identity with respect to "measuring" to some extend identity and privacy. While D13.6 has been focusing on the very modeling approaches from a technical and formal perspective, here we go a step further.

In the present deliverable, we focus on the applicability of privacy models and review as well as illustrate the applicability of models from Deliverable D13.6 using a real-world example. Besides, we show some shortcomings of the approaches presented in D13.6 and include the aspects of combination of information and of misinformation, i.e., information which (partly) cannot to some extend and for some reason be verified by an adversary, hence approaches which may potentially be of major influence in the computation of a measure of anonymity.

Deliverable D13.9 will then go even further and consider large scenarios with respect to quality of results of the identities and anonymity approaches presented here.

Contents

1 Introduction

The central aspect of FIDIS Workpackage 13 *Privacy fundamentals* is to deal with fundamental issues of privacy in relation to identity. Within this workpackage, Deliverable D13.6 [KM07], provided a comprehensive insight into privacy (and context) modelling approaches, namely into technical and formal approaches to privacy quantification. Several different concepts were presented and discussed, based also on Deliverable D13.1 [CM07].

This deliverable must be seen in the light of a series of three deliverables within Workpackage 13, namely D13.6 *Privacy Modelling and Identity*, this one D13.8, and the forthcoming D13.9 *Estimating quality of identity*, which develop the subject of privacy and identity with respect to "measuring" to some extent identity and privacy. While D13.6 has been focusing on the very modelling approaches from a technical and formal perspective, here we go a step further.

In the present deliverable, we focus on four aspects, i.e., on the applicability of privacy models and review as well as illustrate the applicability of models from Deliverable D13.6 using a real-world example. Besides, we show some shortcomings of the approaches presented in D13.6 and include the aspects of combination of information and of misinformation, i.e., information which (partly) cannot to some extent and for some reason be verified by an adversary, hence approaches which may potentially be of major influence in the computation of a measure of anonymity.

Deliverable D13.9 will then go even further and consider large scenarios with respect to quality of results of the identities and anonymity approaches presented here.

In Section 2, we start with re-considering some of the concepts from D3.16 in order to show and extend some shortcomings of the presented approaches, especially of the Common Criteria. Using small "academic" examples we present results which show the (non-)applicability of the formal transcription, i.e., the lacks of the approaches. We especially focus on formal transcriptions of anonymity, unlinkability and unobservability and show the problems arising with the "$\frac{1}{2}$-probability" criterion.

Section 3 focuses on the situation where users belonging to a social network communicate using anonymous messages, hence on the analysis of anonymous communication systems. Information coming from different sources is assumed to be available and must hence be combined, and in this section we focus on a Bayesian approach for computing the combination, feasible typically by a passive adversary observing in- and outputs of the anonymous communication network. Instead of basing the analysis on assuming a uniform selection of communication partners, the analysis is based on a more practical approach of a social network.

The modelization of user behaviour based on behavioural pattern is the focus of Section 4, where histograms are used to cluster and identify sets of users having similar types of communication frequencies. A major issue is the fact that in the case of user profiling, input data is huge and clever preprocessing is indispensable in order to down-size this

input data, such that profiling algorithms can be used afterwards on the preprocessed data. Then, different possible computations of so-called similarities based on this data are presented.

Section 5 shows then the effects of misinformation on the information-theoretic privacy metrics, i.e., what happens if an adversary is not able to verify (all) data. Results are presented for different metrics discussed in D13.6 [KM07].

2 Models

2.1 Introduction

In [KM07] several metrics for determining the extent of anonymity in a situation, for a communication, etc. have been discussed. We will review some of them here (essentially from the Sections 3 and 5) and present some of their advantages but also their problems. For that purpose, we will using small "academic" scenarios. Note that typically, a metric is designed for a special type of situation, e.g., some metrics are designed for global statements about anonymity with respect to set of senders and receivers and the respective messages, whereas others for local statement focused for example on "my" anonymity, etc.

The main purpose of this section is to raise awareness of the problems one might encounter when applying "naïvely" anonymity measures, which eventually are not applicable to the specific situation or which may not be applicable at all. Clearly, the goal is not to elect the one best anonymity measure, but to focus on the properties of the presented approaches. In real world applications other problems might arise, or on the other hand some mentioned problems might not be relevant anymore.

In this section, we will partially assume that the reader is familiar with the concepts and notation used in [KM07] in order to avoid repeating all concepts.

2.2 Common Criteria revisited: Applicability

In Deliverable 13.6 [KM07], Section 3.3 *Common Criteria revisited* we find an attempt to give a formal transcription of existential definitions of CC privacy families. In this section, we want to discuss the (non-)applicability of this formal transcription.

Consider a situation as described in Deliverable 13.6 [KM07], Section 2.4.

2.2.1 Formal transcription in Deliverable 13.6

The text of this whole subsection is cited from D13.6 [KM07].

"The proposal for the CC model privacy formalization is based on the following graphical representation (fig. 1). The set S represents observations of uses of services or resources, P_{ID} is equivalent of subjects and ID stands for users as defined in the CC. Sets U_S and U_{ID} are sets of all possible service use observations and identities, respectively — not only those relevant for a given system. By stating *with probability not significantly greater than* in the following definitions, we mean negligible difference (lower than ε) from a specified value. Let $\mathcal{A}$ be any attacker with unbounded computing power.

Unobservability — there is a space of encodings (U_S) from which some elements are defined to encode use of service/resource (S). However, $\mathcal{A}$ is not able to determine

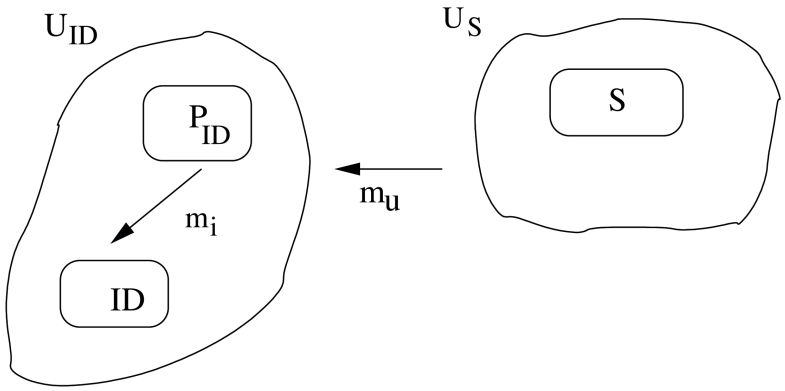

Figure 1: Schematics for the CC view of privacy.

$\forall s \in S$ with a probability significantly greater than 1/2 whether a particular $s \in S$ or $s \in (U_S \setminus S)$.

Anonymity — there is a probability mapping $m_u : S \rightarrow U_{ID}$. When

1. $\mathcal{A}$ knows the set ID — then $\forall\ s \in S, u_{ID} \in ID$, she can only find $m_u(s) = u_{ID}$ with a probability not significantly greater than $1/|ID|$.
2. $\mathcal{A}$ does not know anything about ID (particular elements or size) — then for $\forall\ u_{ID} \in U_{ID}$, she cannot even guess whether $u_{ID} \in ID$ with a probability significantly greater than 1/2. (The probability of finding $m_u(s) = u_{ID}$ would not be significantly greater than 0.)

Unlinkability — let us assume there is a function $\delta : m \times S \times S \rightarrow [no, yes]$. This function determines whether two service uses were invoked by the same $u_{ID} \in U_{ID}$ or not. Parameter m stands for a function that maps service uses (S) into a set of identities U_{ID} (e.g., m_u from fig. 1).
It is infeasible for $\mathcal{A}$ with any δ and any $s_1, s_2 \in S,\ s_1 \neq s_2$ to determine whether $m(s_1) = m(s_2)$ with a probability significantly greater than 1/2.

Pseudonymity — an unambiguous mapping $m_u(s) = u,\ \forall\, s \in S,\ u \in P_{ID}$ exists and is known to $\mathcal{A}$. We assume that there also exists a mapping $m_i(u) = u_{ID},\ \forall\, u \in P_{ID},\ u_{ID} \in ID$, but it is subject to strict conditions and not known to $\mathcal{A}$. When $\mathcal{A}$

1. knows ID, she cannot determine correct u_{ID} with a probability significantly greater than $1/|ID|$;
2. does not know ID, she can only guess with a probability not significantly greater than 1/2 whether $u_{ID} \in ID$."

2.2.2 Interpretation and Applicability

In the sequel, we build academic counter-examples to prove the non-applicability of this formal transcription for anonymity, unlinkability and unobservability. In particular, the probabilistic conditions based on the "$\frac{1}{2}$-probability criterion" are not relevant. This criterion needs to be refined if we want to have such a quantitative criteria in the definitions.

2.2.3 Anonymity

We first consider the anonymity case and, as an example, an authentication protocol based on a challenge-response. During the authentication process, the user gives the answer to a challenge provided by the service, resource, etc. one is trying to authenticate at. We suppose that the right answer belongs to a set of p possible equiprobable answers; only one answer is correct.

1. The legitimate user gives the right answer with probability 1.
2. Any other user gives the right answer with probability $\frac{1}{p}$.

As a first example, we consider a set of two identified users, a legitimate one (u_1) and a non-legitimate one (u_2):

$$ID = \{u_1, u_2\} \tag{1}$$

One of them starts an authentication process. The observer $\mathcal{A}$ knows the set ID; however, she does not know which one of u_1 or u_2 started the process. We suppose that for $\mathcal{A}$ the *a priori* probability for this user to be u_1 (the legitimate user) is the same as the *a priori* probability for this user to be u_2 (the non-legitimate user), i.e., these *a priori* probabilities are $\frac{1}{2}$. In other words,

$$P(u_1) = P(u_2) = \frac{1}{2}. \tag{2}$$

Moreover, $\mathcal{A}$ observes one successful authentication[1], i.e.,

$$S = \{s\}. \tag{3}$$

We clearly have a certain level of anonymity in this process, since $\mathcal{A}$ cannot surely identify the user within ID, the anonymity-set. In Deliverable 13.6, it is stated that in order to have anonymity in the case where $\mathcal{A}$ knows ID, for every $s \in S$ and $u_{ID} \in ID$, $\mathcal{A}$ can only find $m_u(s) = u_{ID}$ with a probability not significantly greater than $1/|ID|$.

In our example, there is only one s and we have $|ID| = 2$. In order to have anonymity according to this definition, $\mathcal{A}$ can only find $m_u(s) = u_{ID}$, i.e., the identity of the user, with a probability not significantly greater than $\frac{1}{2}$.

[1] A correct answer to the challenge is given by the user during the authentication process.

Let us calculate the *a posteriori* probability in our example, i.e., after the observation s, a successful authentication process:

$$P(u_j|s) = \frac{P(s|u_j)P(u_j)}{P(s|u_1)P(u_1) + P(s|u_2)P(u_2)}, \tag{4}$$

where

- $P(s|u_1) = 1$,
- $P(s|u_2) = \frac{1}{p}$,
- $P(u_1) = P(u_2) = \frac{1}{2}$.

Therefore,

$$P(u_1|s) = \frac{1}{1 + \frac{1}{p}} = \frac{p}{p+1} \tag{5}$$

and

$$P(u_2|s) = \frac{\frac{1}{p}}{1 + \frac{1}{p}} = \frac{1}{p+1}. \tag{6}$$

If $p \geq 2$, i.e., if there are at least two possible answers to the challenge, then $P(u_1|s) \geq \frac{2}{3}$. In other words, if $\mathcal{A}$ always chooses $m_u(s) = u_1$, she finds the identity of the user with probability greater than or equal to $\frac{2}{3} > \frac{1}{2}$.

This proves that the condition given for anonymity in the formal transcription of the CC is not relevant. This condition is neither necessary nor fulfilled in general.

Let us investigate further the *a posteriori* entropy of $ID = \{u_1, u_2\}$. This entropy is given by

$$\text{entropy}_{\mathcal{A}}(ID) = \frac{p}{p+1} \log_2 \left(\frac{p+1}{p} \right) + \frac{1}{p+1} \log_2(p+1). \tag{7}$$

This entropy is smaller than 1 if $p \geq 2$, since the probability distribution on this set of two elements is not uniform:

p	1	2	3	4	5	...	2^8	...
$\text{entropy}_{\mathcal{A}}(\{u_1, u_2\})$	1	0.9183	0.81128	0.72193	0.65002	...	0.036753	...

However, it is easy to make the *a posteriori* entropy of ID larger than 1, even when $\mathcal{A}$ finds the identity of the user with probability greater or equal to $\frac{2}{3} > \frac{1}{2}$. Indeed, we can increase the number n of users in ID:

- We consider $n > 2$ users: $ID = \{u_1, u_2, \ldots, u_n\}$.
- We suppose the *a priori* probability distribution to be uniform on ID: $P(u_1) = P(u_2) = \cdots = P(u_n) = \frac{1}{n}$.

- $P(s|u_1) = 1$; u_1 is the legitimate user.
- For every $j \geq 2$ we have $P(s|u_j) = \frac{1}{p}$; u_j are non-legitimate users.

Now,

$$P(u_j|s) = \frac{P(s|u_j)P(u_j)}{P(s|u_1)P(u_1) + P(s|u_2)P(u_2) + \cdots + P(s|u_n)P(u_n)}. \tag{8}$$

Therefore,

$$P(u_1|s) = \frac{1}{1 + (n-1)\frac{1}{p}} = \frac{p}{p+n-1} \tag{9}$$

and for $j \geq 2$

$$P(u_j|s) = \frac{\frac{1}{p}}{1 + (n-1)\frac{1}{p}} = \frac{1}{p+n-1}. \tag{10}$$

The *a posteriori* entropy of $ID = \{u_1, u_2, \ldots, u_n\}$ is given by

$$\text{entropy}_{\mathcal{A}}(ID) = \frac{p}{p+n-1} \log_2 \left(\frac{p+n-1}{p} \right) + \frac{n-1}{p+n-1} \log_2(p+n-1), \tag{11}$$

which is asymptotically close to $\log_2(n)$ for a fixed value of p.

Let us study the case $P(u_1|s) = \frac{2}{3}$. This leads to the condition $n = \frac{p}{2} + 1$ when p is even. The entropy becomes

$$\text{entropy}_{\mathcal{A}}(ID) = \frac{2}{3} \log_2 \left(\frac{3}{2} \right) + \frac{1}{3} \log_2 \left(\frac{3}{2} p \right). \tag{12}$$

In order to get an *a posteriori* entropy larger or equal to x while having a probability of guessing the right user equal to $\frac{2}{3}$, it is sufficient to fulfill the following conditions:

- p, the number of possible answers, is an even number larger or equal to $\frac{8^{(1+x)}}{27}$.
- $n = \frac{p}{2} + 1$

$\text{entropy}_{\mathcal{A}}(ID) \geq$	1	2	3	4	5	...
$p \geq$	4	20	152	1214	9710	...
$n \geq$	3	11	77	608	4856	...

This proves that, even when the anonymity-set has a large entropy, the condition given for anonymity in the formal transcription of the CC is neither necessary, nor fulfilled in general.

2.2.4 Unlinkability

We consider now the unlinkability case. To build a counter-example, we consider a set of $n > 2$ "equiprobable" users

$$ID = \{u_1, u_2, \ldots, u_n\} \tag{13}$$

and a set of $k \geq 2$ observations

$$S = \{s_1, s_2, \ldots, s_k\}. \tag{14}$$

Let us formalize the concept of "equiprobable" users. We suppose that

$$P(m_u(s_k) = u_j \,|\, s_k) = P(u_j) = \frac{1}{n} \qquad \text{for every } j \text{ and every } k. \tag{15}$$

The above condition means that observing s_k does not give any information about who did s_k, i.e., $m_u(s_k)$. This is a perfect example of unlinkability.

According to the formal transcription of CC definitions, in case of unlinkability, it is infeasible for $\mathcal{A}$ with any δ and any $s_1, s_2 \in S,\ s_1 \neq s_2$ to determine whether $m(s_1) = m(s_2)$ with a probability significantly greater than $\frac{1}{2}$.

However, in our example,

$$P(m_u(s_i) = m_u(s_j)) = \frac{1}{n} \qquad \text{if } i \neq j, \tag{16}$$

i.e.

$$P(m_u(s_i) \neq m_u(s_j)) = \frac{n-1}{n} \qquad \text{if } i \neq j. \tag{17}$$

In other words, $\mathcal{A}$ can determine whether $m_u(s_1) = m_u(s_2)$, or not, with probability $\frac{n-1}{n}$ — a probability significantly greater than $\frac{1}{2}$ — by always guessing "no".

This proves that the condition given for unlinkability in the formal transcription of the CC is neither necessary, nor fulfilled in general.

2.2.5 Unobservability

Let us investigate further the revisited definition of unobservability in Deliverable 13.6:

Unobservability — "there is a space of encodings (U_S) from which some elements are defined to encode use of service/resource (S). However, $\mathcal{A}$ is not able to determine $\forall s \in S$ with a probability significantly greater than $\frac{1}{2}$ whether a particular $s \in S$ or $s \in (U_S \setminus S)$." [KM07]

In this definition of unlinkability, the sentence "$\mathcal{A}$ is not able to determine $\forall s \in S \ldots$" needs clarification. From a pure logical point of view, this sentence means that

$$\exists s \in S, \text{such that } \mathcal{A} \text{ is not able to determine } \ldots \tag{18}$$

However, this interpretation would mean, for example, that one undetermined (e.g., unreadable) $s \in U_S$ would make the whole system unobservable, even though for all other elements s_j, the observer $\mathcal{A}$ were able to determine with certainty that $s_j \in S$.

We propose therefore to reformulate the above sentence as follows:

$$\text{"}\forall s \in S,\ \mathcal{A} \text{ is not able to determine } \ldots\text{"} \tag{19}$$

Moreover, the observer $\mathcal{A}$ ***is always able*** to determine $\forall s \in S$ with a probability significantly greater than $\frac{1}{2}$ (even equal to 1) whether a particular $s \in S$ or $s \in (U_S \setminus S)$ by always guessing $s \in S$. We need therefore to reformulate this condition too:

$$\text{"}\forall s \in U_S,\ \mathcal{A} \text{ is not able to determine } \ldots\text{"} \tag{20}$$

We can now build our counter-example. We consider s, an encrypted IP-address of a service (resource) S. Elements in U_S are encrypted blocks of t bits. Let us suppose that the service S has a unique IP address, i.e., $|S| = 1$. The number of elements in U_S — the space of encoding — is $q = 2^t$. We clearly have unobservability, since $\mathcal{A}$ cannot distinguish the encrypted value of the IP address from any other encoded value in U_S.

However, when t is large,

$$P(s \in (U_S \setminus S)) = \frac{q-1}{q} = \frac{2^t - 1}{2^t} >> \frac{1}{2}. \tag{21}$$

The observer $\mathcal{A}$ ***is able*** to determine $\forall s \in U_S$ with a probability significantly greater than $\frac{1}{2}$ (even close to 1) whether a particular $s \in S$ or $s \in (U_S \setminus S)$ by always guessing $s \in (U_S \setminus S)$.

This proves that the condition given for unobservability in the formal transcription of the CC, even after reformulation, is neither necessary, nor fulfilled in general.

2.3 Global notion of anonymity

Consider the entropy based concept[2] which focuses on giving a global measurement of (sender) anonymity. Here, we are in the context of messages sent from an (unknown) sender to a receiver and the attacker tries to determine which one is the sender, hence we focus on sender anonymity, but clearly, the same approach is possible in the "opposite"

[2]Presented in Section 5.3 [KM07], following [DSCP02].

direction (receiver anonymity). In order to stress the shortcomings already mentioned in [KM07, Section 5.4] we reconsider here an overview of some examples given in [THV04].

One of the advantages of the approach is clearly that the computation of the entropy part

$$H(X) = -\sum_i p_i \log_2(p_i) \tag{22}$$

and — in the case of normalization — also the maximum possible entropy, typically computable from an assumed equiprobable distribution of the elements, are quite easy. This is shown in [DSCP02] using the metrics for simple mix networks crowds, and onion routing.

On the other hand — as already noted by the authors — the determination of the probabilities is the difficult part.

In order to present possible problems, consider now the two situations presented in [THV04], when a message is coming from a set of n senders. We assume that the attacker is able to determine the probability distribution on the set of senders for this given message.[3]

- On the one hand, consider 10 possible senders and assume the attacker has determined uniform distribution among these ten, i.e. $p_i = \frac{1}{10}$ for $1 \leq i \leq 10$
- On the other hand, consider 26 possible senders with $p_1 = 0.5$ and $p_i = \frac{0.5}{25}$ for $2 \leq i \leq 26$.

For both cases, we have — when considering unnormalized entropy — the same result, i.e.

$$S = -\sum_i p_i \log_2(p_i) = 3.3219 \tag{23}$$

This is just one possible simple example, others can be constructed as well (cf. [THV04]).

From the local perspective, i.e., when the anonymity of a specific sender (or receiver) is of interest, clearly the two situations appear as not being identical with respect to anonymity as in the first one, there is no sender which is more probable than the other ones, while in the second one, the difference of probabilities between them is huge (one has probability 0.5, the other ones only 0.02 each).

However, the measurement above (23) indicates that from the global point of view of the approach for measuring anonymity presented above, i.e., when the anonymity of the whole set of (possibly) communicating parties is concerned, both situations should be considered as equally anonymous. Tóth et al. [THV04] indicate that here, what is measured is really the "amount of information that is required to break *totally* the anonymity of the message." In practice we are however typically more interested of

[3] For details of these models please see [THV04].

determining a link between some given message(s) and a person with high probability, which is quite another type of problem.

In the case of normalized entropy, the situation is similar, the same issues may arise. Normalized entropy here is defined as

$$d = \frac{H(X)}{H_M} \tag{24}$$

where H_M is the maximal possible entropy, typically the logarithm to the base two of the size of the set of senders. The resulting anonymity factor is therefore contained in the interval $[0, 1]$ (assuming $d = 1$ if only one sender is present).

Tóth et al. present — like for the case of unnormalized entropy above — two different situations resulting in the same value $d = 0.7067$:

- On the one hand, consider 25 possible senders with $p_1 = 0.5$ and $p_i = \frac{0.5}{24}$ for $2 \leq i \leq 26$.
- On the other hand, consider again 25 possible senders with

$$p_i = \begin{cases} 0.16644 & 1 \leq i \leq 5 \\ 0.00799 & 6 \leq i \leq 25 \end{cases} \tag{25}$$

In the first situation there is one sender with a probability (0.5) which is much larger than all other ones (0.021), whereas in the second situation, each one of the five senders has a much higher probability than each one of the 19 other ones. Again, the same remarks as indicated above in the context of unnormalized entropy apply in this case of normalized entropy: different local situations of anonymity are measured as being identical from this global measurement of anonymity.

In Section 5, this approach as well as many others are re-considered with respect to the problem of mis-information.

2.4 Conclusion

We have shown how the applications of common criteria might raise problematic issues. Academic counter examples prove that the probabilistic conditions based on the "$\frac{1}{2}$-probability criterion" are not relevant:. First, we have presented a counterexample to the case of "anonymity" which shows that the presented application of common criteria is based on a condition which is neither necessary nor fulfilled in general. The counterexample created respects anonymity according to the transcription of the definition, yet we show that the attacker has a possibility to find the identity of the user with a probability of at least $\frac{2}{3}$ (in non-trivial cases).

Second, counterexamples for the case of "unlinkability" and "unobservability" show that they also are formulated based on conditions which are neither necessary not fulfilled in

general. Note that all these conditions mentioned depend heavily on the "$\frac{1}{2}$-probability criterion", which seems to be the central problematic issue here.

A next step would consist of having a critical look at all other measures presented so far, and eventually model situations where a measure is applicable and under which explicit and exact premises. This does not at all mean that we want to disparage such measures in general, one of our our main concerns is to raise awareness that the application of measures within this context is not an easy issue, lots of measures are available, but not all will fit and give the appropriate results in a specific situation.

3 Using Bayesian inference to combine several sources of information

3.1 Introduction

In the last few years defining and quantifying anonymity in the context of communication networks has been a hot research topic. A substantial set of papers focus on the definition of anonymity, others present designs and analysis of new anonymous communication systems or attacks of existing ones. Yet, more focus on the theory of mix systems is needed in order to improve our fundamental understanding of anonymity properties which are possible or practically achievable.

We consider the anonymity of users belonging to a social network who communicate to 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.

This work belongs to a growing body of research focusing on the anonymity analysis of anonymous communication systems. A substantial part of this literature consists of papers evaluating the effectiveness of mix-based anonymity systems in a theoretical setting; e.g., [Dan04, DS03, SDS02]. Such work often involves assumptions such as "users pick their communication partners uniformly at random" which help with the mathematics of calculating anonymity, and hence aid our understanding and intuition, but do not necessarily hold in practice. Furthermore, the authors often examine properties of the anonymous communication systems and shy away from incorporating models of users. We take a more practical approach by assuming a social network, deriving the attacker's knowledge about users based on the fact that they belong to such a network and then evaluating the anonymous communication system in the context of this knowledge.

In order to evaluate anonymity in a practical setting, it is necessary to incorporate a priori information the attacker might have about communication patterns of users. We briefly mention a number of papers that explore related research problems. Sweeney [Swe02] proposed the *k-anonymity* model to quantify the anonymity of user records in a database, given that some of the user attribute values are available to the adversary. Diaz et al. [DCSP02] assume that some information on user properties is known, such that the user base can be partitioned in different groups that share a similar profile. Clauß et al. [Cla06, CS06] propose a framework and metrics for systems where the adversary has some information on user attributes. In these papers the focus is on user properties or profiles, and little effort is made to combine the knowledge gained through traffic analysis with the profile information available to the attacker. In [Cla06, CS06], it is mentioned that the communication layer information gained through traffic analysis can be modelled by means of attributes, but no concrete example is given of how this could be realized. Finally, Diaz et al. [DTD07] showed a toy example where the combination of user sending profiles and data gathered through traffic analysis resulted in higher anonymity, contradicting what had been claimed in [CS06]. However, no gen-

eral methodology was given in [DTD07] for computing anonymity metrics when several sources of information are available. The most closely related paper which attempts to combine knowledge about profiles with traffic analysis information is [DS04] where a lot of the Bayesian theory we use is presented, but only a brief demonstration of the technique is given.

Perhaps the most related piece of related work in terms of the spirit of the analysis and in the style of the results obtained is one of Dingledine and Matthewson [MD04]. They employ simulations in order to evaluate the effectiveness of statistical disclosure attacks on a model of an anonymity system; i.e., they attempt to recover profiles from the communications data while we build assumptions about profiles from the social network and then add the communications data on top.

3.2 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 modelled as a mix[4]. Since Chaum [Cha81] first proposed mixes for achieving anonymous communication in 1981, multiple designs have been proposed in the literature both for low-latency communication, e.g. [DMS04] and for high-latency, message-based communication [Cot, DDM03, KEB98].

The adversary we consider can observe all input messages arriving at the mix (and their respective senders), as well as all output messages leaving the mix (and their recipients), but not the internal operations of the mix. Naturally, the messages are encrypted so the content is hidden. Although the attacker does not know the exact 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 behaviour. 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." For convenience we have assumed that user profiles are static. Extending the analysis to dynamic profiles (i.e., profiles that change over time) would be simple (for a message sent at time t_0, one just needs to use the probabilities that correspond to the profile at time t_0). The main challenge is however, to create a mechanism that adaptively finds the dynamic profiles. This is left as a subject of future research.

3.3 Anonymity with one source of information

We draw on the literature, more specifically [DSCP02] and [SD02], for our definition of anonymity. The basic idea of these metrics is to use the Shannon entropy [Sha48] of the probability distribution linking subjects to a message or action (normalized entropy in

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

the case of [DSCP02]). The entropy of this probability distribution gives a measure of the uncertainty concerning the identity of the subject who originated/received a message. Entropy-based anonymity metrics take into account both the number of users in the system and their probabilities of being linked to a particular action, and anonymity increases both with the number of users and the uniformity of the probability distribution linking them to messages.

The goal of our adversary is to identify the recipient of messages arriving at the mix (*recipient anonymity*) or the sender of messages leaving it (*sender anonymity*). Therefore, the adversary makes hypotheses of the type "hypothesis h_j is true if u_j is the sender (recipient) of this outgoing (incoming) message," and computes the probability $\Pr(h_j)$ that h_j is true. Given that every message has one sender and one recipient, the probabilities $\Pr(h_j)$ sum to one (i.e., $\sum_{j=1}^{N} \Pr(h_j) = 1$).

Here we use the *effective anonymity set size* [SD02] as the metric for sender and recipient anonymity. For a given message entering (leaving) the mix, the recipient (sender) anonymity A is given by the Shannon entropy of the probability distribution of each of the hypotheses h_j being true; i.e., $A = -\sum_{j=1}^{N} \Pr(h_j) \log_2(\Pr(h_j))$.

Let us first illustrate how anonymity is computed when only one source of information is available to the attacker. If the attacker knows the sending profiles of users, but cannot observe the inputs and outputs of the mix, the recipient anonymity of a message sent by user u belonging to the user population U is given by the entropy of her sending profile. That is, if u chooses user u_j as her recipient with probability $\Pr(u \rightarrow u_j)$, then the recipient anonymity provided by u's profile is $A_p = -\sum_{j=1}^{N} \Pr(u \rightarrow u_j) \log_2(\Pr(u \rightarrow u_j))$. Conversely, when u receives a message, the anonymity of the sender is given by $A_p = -\sum_{j=1}^{N} \Pr(u \leftarrow u_j) \log_2(\Pr(u \leftarrow u_j))$, where $\Pr(u \leftarrow u_j) = \frac{\Pr(u_j \rightarrow u)}{\sum_{k=1}^{N} \Pr(u_k \rightarrow u)}$ is the probability of u_j being the sender of a message received by u. In the remainder, we denote the sending profile of a user u as $P(u \rightarrow U) := \{\Pr(u \rightarrow u_j), \forall u_j \in U\}$ and its recipient profile as $P(u \leftarrow U) := \{\Pr(u \leftarrow u_j), \forall u_j \in U\}$.

Alternatively, we can consider an adversary who can see the inputs/outputs of the mix but does not have a priori knowledge of user profiles. The probability of an input (output) message matching each of the outputs (inputs) depends on the type of mix, overall traffic load and the timing of messages. Let us consider a timed pool mix. Pool mixes work in cycles called *rounds* that comprise three steps (1) **collect:** it collects messages from senders for a period of time T; (2) **store:** upon being received, messages are decrypted with the mix's private key (which allows it to retrieve the destination address), and stored in an internal memory called *pool*; and (3) **flush:** once the timeout T has expired, a fraction of the messages are randomly selected and sent to their destinations, while the rest is kept in the pool for the next round.

The probabilities of matching the mix inputs and outputs are computed as follows [DP04]. Let m_r be the number of messages contained in the mix in round r (prior to the mix flushing), and s_r be the number of messages sent by the mix in round r. If

a message M arrived at the mix in round r, its probability $\Pr(M = O_{r',i})$ of matching each of the $s_{r'}$ outputs $O_{r',i}$ that left the mix in round r' is:

$$\Pr(M = O_{r',i}) = \begin{cases} 0 & \text{if } r' < r \\ \dfrac{1}{m_{r'}} & \text{if } r' = r \\ \dfrac{1}{m_{r'}} \displaystyle\prod_{k=r}^{r'-1} (1 - \frac{s_k}{m_k}) & \text{if } r' > r \end{cases} \tag{26}$$

The recipient anonymity A_m provided by the mix to message M is given by the entropy of the probabilities $\Pr(M = O_{r',i})$. The computation of the probabilities $\Pr(I_{r',i} = M)$ linking an output M to all possible inputs $I_{r',i}$ is analogous, and their detailed derivation can be found in [DP04]. Note that probabilistic relationships between inputs and outputs can also be derived for other types of mixes such as Stop-and-Go [KEB98].

3.4 Anonymity with several sources of information

Bayesian inference is an approach to statistics in which all forms of uncertainty are expressed in terms of probability. It starts with an initial set of beliefs represented by an a priori probability distribution, which is updated as new evidence is collected. The distribution indicates how likely it is for a hypothesis to be true.

Let h_j be the hypothesis that user u_j is the sender (or recipient) of a given message received (or sent) by user u, and $\Pr(h_j)$ the prior probability of this hypothesis being true. Let E be some evidence or observation that gives 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 $\Pr(h_j|E)$ of h_j, given that we have obtained evidence E. We denote this probability distribution by $P(H|E) = \{\Pr(h_j|E), 1 \leq j \leq N\}$:

$$\Pr(h_j|E) = \frac{\Pr(h_j)\Pr(E|h_j)}{\sum_{k=1}^{N} \Pr(h_k)\Pr(E|h_k)} \tag{27}$$

In our setting, we consider that both sender profiles and mix input/output observations are available to the adversary. The prior probability $\Pr(h_j)$ is given by the sending profiles of users, and corresponds to $\Pr(a \rightarrow u_j)$ in the case of recipient anonymity, and to $\Pr(a \leftarrow u_j)$ for sender anonymity (as explained in the previous section). The conditional probability $\Pr(E|h_j)$ is computed as follows. For recipient anonymity (analogously for sender anonymity), let R_j be the set of messages received by user u_j. Given that u sent message M to u_j (i.e., h_j is true), the probability $\Pr(E|h_j)$ of observing the evidence E corresponds to the probability of the mix matching M to one of the messages received by u_j:

$$\Pr(E|h_j) = \sum_{O_{r',i} \in R_j} \Pr(M = O_{r',i}) \tag{28}$$

Bayesian inference can be applied recursively if new independent evidence E' becomes available to the adversary.

3.5 Example of application

Let us consider that user Alice has a known profile of communicating with two other users (u_0 and u_1) with a probability distribution $\Pr(a \rightarrow u_j)$ such that:

$$\Pr(a \rightarrow u_0) = 0.4$$
$$\Pr(a \rightarrow u_1) = 0.6$$

We consider that Alice sends a message $\mathcal{M}$ through a mix to a receiver u_j chosen according to the distribution $\Pr(h_j) = \Pr(a \rightarrow u_j)$. Based on Alice's profile information only, the anonymity $\mathcal{M}$'s recipient is:

$$A = -\sum_{j=0}^{1} \Pr(a \rightarrow u_j) \log_2 \Pr(a \rightarrow u_j) = 0.97 \ . \tag{29}$$

Let us assume that Alice sends her messages through a threshold pool mix with threshold $T = 2$ and pool $P = 1$. For this mix, $m_k = 2$ and $s_k = 1$ for every k, and there is only one message per round r:

$$\Pr(M = O_{r',i}) = \begin{cases} 0 & \text{if } r' < r \\ \left(\frac{1}{2}\right)^{r'-r+1} & \text{if } r' \geq r \end{cases} \tag{30}$$

In order to further identify the recipient of $\mathcal{M}$, the adversary monitors the outputs of the mix from the moment t_0 when $\mathcal{M}$ was sent by Alice, and sees that the first output goes to u_0 and the fourth goes to u_1. For simplicity, we assume that the other recipients are people to whom Alice never sends messages. Taking into consideration the mix only, the adversary derives the probability $\Pr(E|h_j)$ as indicated in Equation 28, which gives as result:

$$\Pr(E|h_0) = \frac{1}{2}, \qquad \Pr(E|h_1) = \left(\frac{1}{2}\right)^4$$

Thus, the combined anonymity that takes into account both the profile and the mix input/output observations is:

$$\Pr(h_0|E) = \frac{\Pr(h_0)\Pr(E|h_0)}{\sum_{j=0}^{1}\Pr(h_j)\Pr(E|h_j)} = \frac{0.4 \cdot 0.5}{0.4 \cdot 0.5 + 0.6 \cdot 0.5^4} = 0.84$$

$$\Pr(h_1|E) = \frac{\Pr(h_1)\Pr(E|h_1)}{\sum_{j=0}^{1}\Pr(h_j)\Pr(E|h_j)} = \frac{0.6 \cdot 0.5^4}{0.4 \cdot 0.5 + 0.6 \cdot 0.5^4} = 0.16$$

And the resulting anonymity for message $\mathcal{M}$ is given by:

$$A = -\sum_{j=0}^{1}\Pr(h_j|E)\log_2\Pr(h_j|E) = 0.63 \ .$$

3.6 Conclusion

In this section we examined the anonymity of users in the practical context of a social network. We proposed a Bayesian method to combine several sources of information and show how it can be applied to compute anonymity when the adversary has access to both profile information and anonymous network traffic observations. Although in our examples we have focussed on combining two sources of information, we note that Bayesian inference can be applied recursively, should new sources of information become available.

The end result of applying Bayesian inference is a probability distribution that indicates the likely senders or receivers of messages. The entropy of this distribution provides a metric for sender and recipient anonymity with all available information taken into account.

4 Data preparation and user profiling

This work follows-up previous Deliverable D13.6 [KM07] where an overview of selected modelling approaches was provided. This section presents current work on traffic log processing and user profiling. The aim is to find an approach to model user behaviour based on behavioural patterns. Since the amount of input data this process involves is really large, effective preprocessing is crucial for the profiling to provide significant results. This section presents our approach to restricting the input data with respect to its relevance. We use histogram clustering to identify sets of users with similar frequencies of communication; entropy and TF-IDF (Term Frequency – Inverse Document Frequency) help to select destinations that are relevant for a given set of users; the PrefixSpan algorithm is used to find frequent communication patterns. The main profiling is done with preprocessed data and our experiments show that the proposed approach to restricting the input has a positive impact on the significance of results. At the end of this section we discuss our methods of calculating similarities between users to estimate how similar their behavioural characteristics are.

4.1 Introduction

User profiling is a technique that (mainly online) services use to provide customizable content to their users. These services operate with a *user profile* that reflects users' behaviour in the past. This is the type of information that customizable content should be based on. Profiling gives us some evidence about users' behaviour (information about their interests and social background). Instead of requiring users to state their interests, a service could learn this information from users' past activities.

We can learn a lot about an individual based on his *context* and we can also predict possible future actions by combining several types of context information in an efficient way [KM07]. This can reveal some *private* data about users and therefore privacy protection issues must be carefully taken into consideration. A critical term here is a *user profile*. If we have a good profile of a user (his behaviour) we can try to find and link together similar profiles and, with some probability, identify a user — in this case, of course, we do not know the real identification of a user, we just know that a given set of profiles exhibits some similarities. Accuracy of such a model is a key feature for effective predictions of users' future actions that are based on this model.

The main question is how much private information can be derived from context information that is related to an individual. The second question is to what extent a behaviour pattern can be used to pinpoint an individual among others. Discovering valuable information in huge databases involves several phases such as data preprocessing and cleaning; data transformation into some common structure; data mining techniques to identify interesting patterns or correlations among parts of input data; interpretation of the results such as visualization or some kind of user friendly format.

In this section we would like to focus on the data preprocessing (and partially profiling) phase, which is one of the key issues for the whole process to be successful. We will introduce our approaches of working with the data and the restrictions that we use to limit the input data in a reasonable way. We will also discuss how efficient these restrictions are. Our expectations are that these restriction rules will help to optimize the input data in terms of relevance and help to focus on data that has some interrelations. One such technique is to split data into different "levels" according to some characteristics and process these levels individually. This helps to concentrate on specific pieces of data only. We will describe our techniques for identifying different levels of activity and different levels of relevance.

Filtered data forms vectors of behaviour that are used as the input for the profiling. Again, we can use a clustering method (to find sets of users whose vectors of behaviour exhibit some similarities), but we also want to try to process the data with the PATS model (Privacy Across The Street) [MC04], which aims to incorporate all relevant available context information in order to find how much private information can be inferred. At the end of this section a metric is presented that we use to calculate similarities.

4.2 Input data

Significant user profiling cannot be successful with small amounts of input data. We have data from the global IP traffic log of the university network (NetFlow). This log collects information about all communication in the network as well as with the outside world. For every connection, both source and destination IPs are stored, source and destination ports and the amount of data transmitted. The tables are created every hour and at the end of each day all hourly tables get aggregated into daily tables. Daily tables are further aggregated into weekly and finally monthly tables. On average, the daily load is around 150 000 000 rows in hourly tables. In the aggregation process irrelevant data (like SNMP and other management-based communication) is dropped to decrease the number of rows and multiple rows related to each individual communication are put together. Daily tables consist of 3 000 000 rows on average. The whole dataset is stored in a MySQL database on a dedicated machine.

We agreed with our IT department, who provided us with the data, not to disclose any kind of information that would help to disclose real person identity or behavioural characteristics. Therefore all real IP addresses (both the source and the destination) were covered up.

To cope with this extremely large input data in an efficient way, we have to deploy sets of restrictions that let us work with focused pieces of data. It would not be reasonable to take the whole database as such and start with some profiling techniques, since the output would not be descriptive in any way. A more natural way is to split the data into some relevant subsets.

Data selection and optimization is crucial to get a significant output that can be used for predictions of users' future behaviour. There has always been a trade-off between

the size of the input data and its actual relevance. On the one hand, the more data we use the more accurate the results we get (in terms of predictions of future actions). On the other hand, to get clearly bounded results one would start decreasing the amount of data while keeping the relevance as high as possible. We use cluster analysis to find sets of users with similar behaviour tendencies and therefore we are aiming for significant results in terms of the clustering process (i.e. reasonable numbers of bounded clusters). Users in these clusters can be explored with additional input data to find out more detailed information about actual differences in behavioural characteristics.

4.3 Data preparation

In order to get conclusive results out of the final profiling process, the data preparation must be carried out with a great care. This section is dedicated to the description of our current approach, reporting interesting findings and discussing deployed mechanisms.

Source IP addresses, destination IP addresses, destination port and the number of connections that were established during a particular period of time (day(s), week(s) or month(s)) is important information from the database. Putting this together, we can create a two-dimensional matrix, where each column is one destination IP 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 i to the destination IP j using a given destination port (or a set of destination ports). Each row of this matrix reflects the communication pattern of a particular source IP (we can call it a vector of source IP behaviour). The idea is to process all these vectors with a clustering (profiling) algorithm to get sets of similar vectors as output.

From the information above and the description of the input data in Section 4.2 one can easily observe that the dimensions of the matrix would be quite large. This is a problem we have to deal with, because clustering (or calculating similarities) these vectors will not give very significant and reasonably interpretable results. The final dendrogram (graphical representation of a clustering process) is very detailed without any bounded clusters indicating separate sets of source IP addresses with similar behavioural characteristics. We performed this graphical representation in order to have an insight into the clustering process and to be able to discuss the appropriateness of further preprocessing steps before the final clustering. From this observation, we realized that with input data that had some prior relations, (e.g., only one department instead of the whole university) the results will be more significant in terms of identification based on behavioural characteristics.

In addition to the matrix dimensions there is another problem. The matrix contains many zeroes and overall is very sparse. So the aim is to decrease the dimensions and try to push the number of zeroes as low as possible.

Before we describe specific approaches to optimize the matrix, we need to mention the very first set of restrictions performed with the input data. In this step we create a new

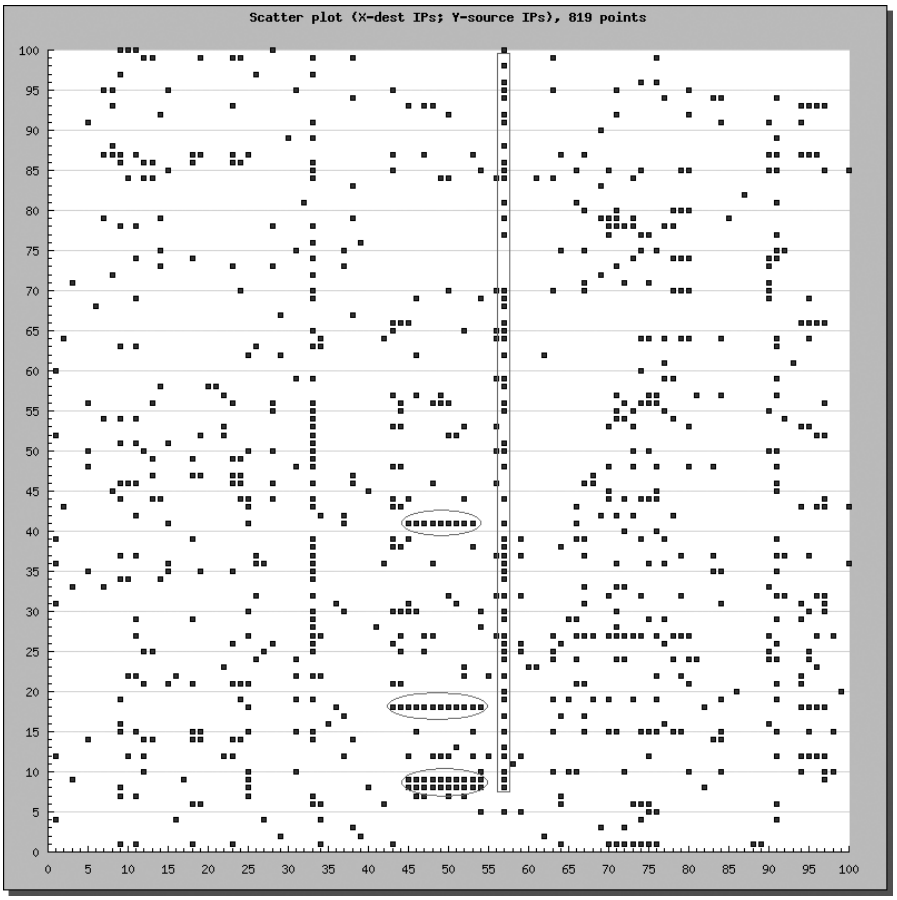

Figure 2: Scatter plot — x-axis represents destination IPs and y-axis represents source IPs; points mean that there was a communication between a source IP to a destination IP (frequencies are not considered). There is an indication of one commonly visited destination as well as some behavioural similarities.

database table that consists of selected source IP ranges and destination ports (e.g., our faculty and ports 22, 80 — `ssh`, `http`). This is done both to decrease the size of the database table and to focus 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 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.

Some kind of visualization should help to get an initial view on the data. We decided to use a simple scatter plot indicating whether there was a communication (no matter how frequent) between any two source and destination IPs. This can help to detect commonly visited destinations and some patterns of similar behaviour (as shown in Figure 2).

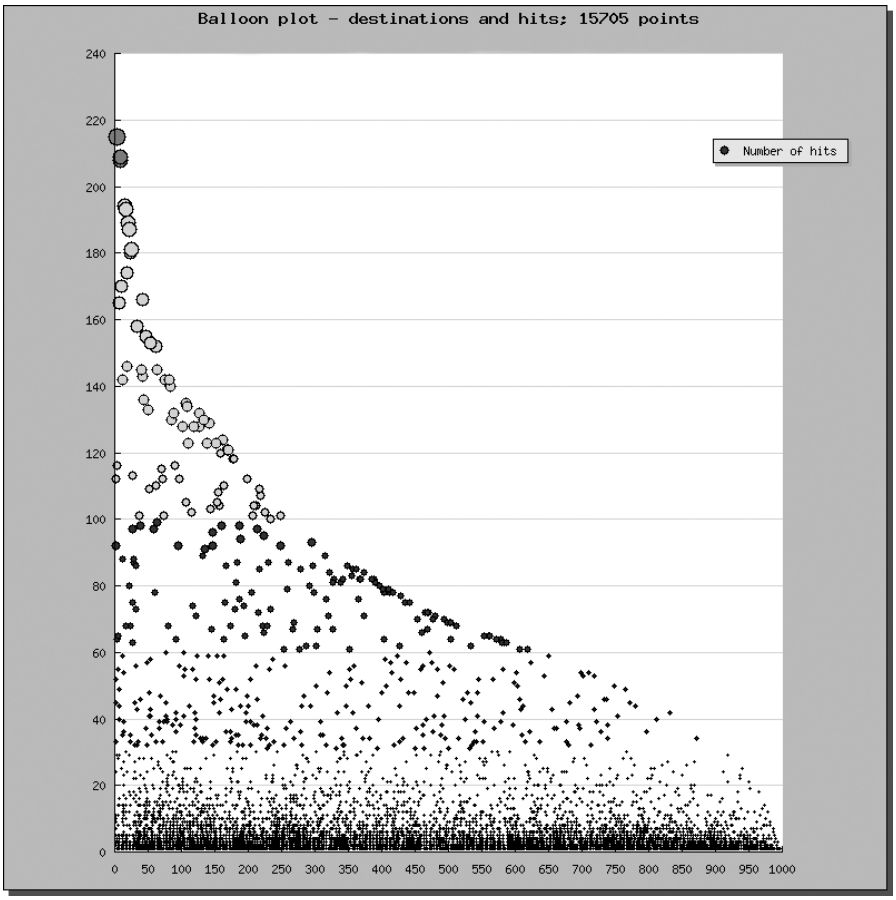

Figure 3: Balloon plot — helps to find optimal upper bound of hits to restrict destinations that are visited very frequently by only one source IP. Y-axis represents number of visits; X-axis is an index of destinations.

To involve frequencies of communication, we use a balloon plot (Figure 3). This helps for the decision of setting an upper (or lower) bound of maximal (or minimal) number of connections allowed. This restriction may help to indicate destinations that are visited very frequently by only one source (or the other sources are very passive). If this frequent communication happens regularly with respect to a selected period of time, we can clearly detect it and therefore there is no need to incorporate these destinations in the matrix. Very passive source IPs on the other hand also bring no additional information into the profiling process and therefore we can get rid of them too. The idea that this restriction is based on is that pinpointing "borders" (in terms of frequencies) of communication and generally splitting the whole dataset into three sets (very active, very passive, the rest) may help to make the matrix of behavioural vectors more dense. And as a consequence this helps the profiling stage to process source IP addresses with behavioural vectors that are as dense as possible.

The optimal number of "hits" (according to our experiments) in case of e.g. `ssh` connections and one month period seems to be around 10.

The initial restriction possibilities described above are very basic. The resulting matrix is still quite large. The data is stored in a new table so we do not need to take account of the computational time since the new table is many times smaller than the original one(s). Working with the original table(s) would be quite hard, because MySQL was not originally intended for such extreme loads (when compared, applying subnet(s) restrictions in the original table(s) took dozens of minutes and since we want to work with this restricted subnet(s) only, this computation would have to be done in each step).

4.3.1 Source IPs optimization

If we want to use profiling to find sets of source IPs with similar behavioural tendencies and if we expect some reasonable level of significance, we need to preprocess the set of source IPs in an efficient way. The idea is that there are several levels of source IPs "activity" — some sources are quite passive in communication, while other sources are extremely active. If there is a set of IPs with similar behaviour, it will possibly exhibit some similarity in frequency of communication. Even though this assumption may not hold for every situation, separating source IPs into different levels of activity can lead to an increase of the accuracy of the consecutive processes.

Our approach to find different levels of source IPs activity is quite straightforward. We build histograms for all sources to describe how many destinations were visited how many times during a given period of time. First we find the maximum number of accesses and then create aggregated intervals of ten accesses. Then for each source IP and each interval the number of visited destinations is found (e.g., source IP A visited two destinations five times). We discovered during our experiments that the majority of records fall into the first category — zero to ten accesses. Therefore we decided to split the first interval into ten groups, each of which represents a specific number of accesses. Figure 4 shows such a histogram. The information from each histogram is stored as a vector to a CSV (comma-separated values) file. It reflects the frequency behaviour of each source IP. This file is processed with a clustering algorithm to find clusters consisting of IPs with similar frequency vectors.

The set of vectors is processed with R [R D06] and Ward's clustering method [War63] is used. Ward's clustering method seems to be the best option for this case since it produces quite sharp and balanced clusters. Let us describe this method in a little more detail.

Ward's clustering procedure is somewhat different from the classical approaches that measure the distances between clusters in a geometric way. It calculates the "information loss" to evaluate the distances between clusters and it aims to minimize this information loss. At each step of the process, all possible cluster pairs are considered and the two clusters whose fusion gives the smallest information loss are combined. Information loss

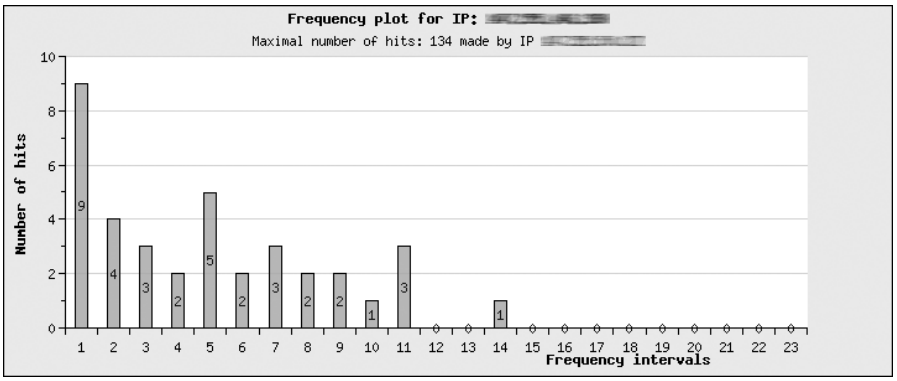

Figure 4: Frequency plot — displays frequencies of accesses. First ten levels are fixed numbers of accesses. The eleventh interval denotes 11-20 accesses; twelfth interval denotes 21-30 and so on.

was defined by Ward in terms of an error sum-of-squares criterion (ESS). As an example, consider six values $(0, 0, 1, 1, 3, 3)$. All values in one cluster will result in ESS $= 9.5 = 2 * (0 - 1.5)^2 + 2 * (1 - 1.5)^2 + 2 * (3 - 1.5)^2$, while three clusters $(0, 0), (1, 1), (3, 3)$ give ESS $= 0$ (1.5 is the mean of values). The benefits of Ward's clustering method include strong tendency to split data into groups of roughly equal size and no clusters with only one or few elements. This method is not efficient for data with large "natural" clusters since it will tend to split them. Our data does not have this property, so we can use this method.

Classical methods such as complete linkage, single linkage and average linkage result in imbalanced dendrograms with no visible clusters. According to our experiments, Ward's method is the best option to find different levels of source IP communication frequencies. An example of a dendrogram produced with Ward's method is in Figure 5.

To process the individual sets separately, we need to know which cluster each source IP belongs to. We use R and the `cutree` method to cut the dendrogram by specifying either the number of cuts we want to get or a height at which a cut will be processed. After this step we obtained distinct sets of source IPs. The distribution of each set for the dendrogram in Figure 5 is shown in Figure 6.

The approach we have just described seems effective for reducing the size of the matrix with regard to the source IP attribute. We concentrate on IPs that show some initial similarity. This helps to produce more significant and conclusive results. This process is only one half of the whole optimization of the matrix. The other part focuses on the set of destinations and aims to decrease the second dimension of the matrix. It also helps to increase its density (minimize the number of zero cells).

Figure 5: Dendrogram with quite clear and distinct clusters that consist of source IPs with similar frequencies of communication.

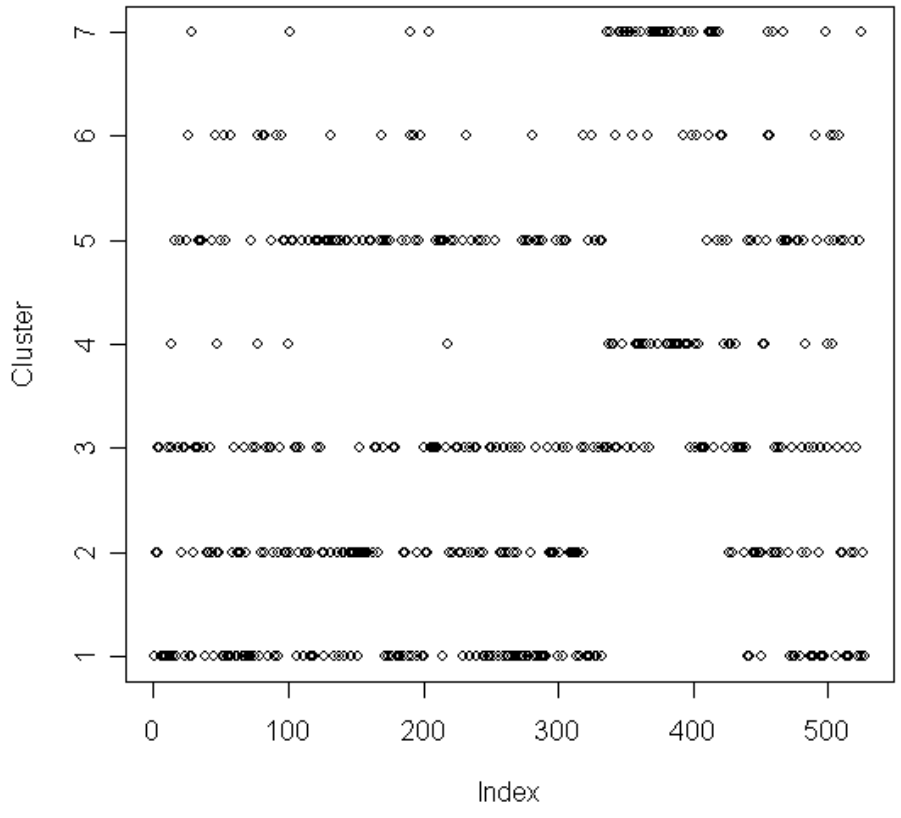

Figure 6: Distribution of source IPs into distinct clusters. Y-axis shows the number of cluster, X-axis is an index of source IPs (this case consisted of 527 source IPs).

4.3.2 Destination IPs optimization

So far we have discussed our approach to source IP address optimization. We also have to take into account the large number of destination IPs. This step is heavily influenced by the very first restriction rules discussed in Section 4.3. The number of destinations depends on which destination port(s) were considered (e.g., 22 – `ssh` vs. 80 – `http`). Based on this fact the destination IPs' optimization should be applied only if needed (contrary to the source IPs' optimization, the use of which is reasonable under any circumstances).

The goal for this optimization is both to decrease the dimensions of the matrix and to increase its density. We need to be able to differentiate between destinations to say how relevant they are for the current set of sources. Based on this knowledge we can drop (or process separately) destinations with little relevance. There are many open questions and problems that arose from our experiments so far. We will discuss these at the end of Section 4.3.2.

We use information entropy to find the distribution of visits to a particular destination. This value is computed and compared with the maximum possible entropy to learn valuable information about each destination. If there is a destination that is visited by almost all sources, its entropy will be close to the maximum entropy. These types of destinations can be dropped since their existence does not bring any new information. On the other hand, if the entropy is zero, we know that there is only one visitor. We could also drop these destinations, but here we need to be more careful. Destinations with only one source and few accesses can be dropped without any significant implications. If many (experiments showed that for `http` this value starts at a few dozen) accesses are made by only one source, we can say that the source is really interested in that destination. This helps a lot to identify a source IP, especially if this interest persists for a longer period of time. Therefore this kind of information is of great importance to us and experiments showed that this situation is not unusual.

Entropy can help to decrease the number of destinations, but it has to be used with care. To be able to process destinations with the same entropy, we slightly changed the way the matrix is constructed. Instead of adopting a frequency-based approach where individual cells contain the number of visits, we adopted a usage-based approach, where individual cells contain a one if there was a communication and a zero otherwise. The idea is to get a smaller number of distinct entropy levels. If we had used a frequency-based approach, destinations with the same number of sources would have had different entropies because of different number of visits. Using a usage-based approach, this difference is removed and destinations fall into the same set. Later we can use these sets and process them individually using the frequency-based approach. Together with source IP optimization we get a matrix where dimensions are significantly lower and the matrix density increases along with increasing entropy.

Inverse Document Frequency There is another technique that can be used to evaluate destination relevance. This approach is very well know in the text mining and information retrieval area and is called TF-IDF (Term Frequency – Inverse Document Frequency) [Ber03, LB04]. TF-IDF is a weight used to evaluate how important a word is to a document in a collection of documents. Inverse document frequency is a measure of the general importance of the term in a set of documents. The formula for TF-IDF is [Ber03]

$$weight(i,j) = tf_{i,j} \cdot \log_2\left(\frac{n}{df_i}\right), \qquad \text{if } tf_{i,j} \geq 1 \tag{31}$$

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

If we consider one destination IP as a term and a vector that describes source IP 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 ($\log_2(\frac{n}{df_i})$), we can generalize 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 a given sets of both source and destination IPs. This "index of importance" (for every destination IP) will vary with every little change of either set of IPs.

Boundaries within which IDF falls range from zero (this is the case where every source visited that destination – $\log_2(\frac{n}{n})$) to $\log_2(\frac{n}{1})$ (this is the case where only one source visited that destination). It can be easily seen that the upper bound $\log_2(n)$ is actually the maximal possible entropy. Entropy and IDF are in negative correlation — with increasing entropy IDF decreases and vice versa.

IDF evaluates destination relevance and relies on the number of sources that visited that destination. The IDF weight will be same for the destinations with the same number of sources. The entropy for destinations with the same number of sources will vary because the probability of a visit is also considered. We can use both facts to identify different "levels" of destinations with more evenly distributed visits. If we sort destinations based on IDF in a descending order and use entropy as a second criterion, we can identify those "levels". For each level of IDF we select the destination with the highest entropy. Using this approach (or a modification of it) we are able to distinguish different "levels of interest" that help to decrease the matrix dimensions and bring more focused information into the profiling process.

Having these types of restrictions rules deployed, we can observe their influence under different circumstances (caused by the first set of restrictions) or during different periods of time. We can also use IDF to select destinations with a bounded number of unique sources.

PrefixSpan If there is a set of users with similar behaviour then the sequence of destinations will be same. Therefore we can consider using a sequential pattern mining algorithm to search for frequent sequences of destinations in the matrix of IPs. With this information we can process these destinations separately, find respective source IPs and observe their behavioural characteristics.

We use PrefixSpan [PHA+01], a sequence mining algorithm which recursively searches for frequent patterns by projecting a sequence database into a set of smaller databases based on frequent *prefix* mined so far. Sequences identified with PrefixSpan do not need to be continuous in the input file. As an example let us assume that the input files contains *a, b, d, e, f, h* as a set of destinations visited by source IP *A* and *c, b, g, e, f* as a set of destinations visited by source *B*. PrefixSpan will find (in addition to *b, f* and *e, f*) two occurrences of a sequence *b, e, f*.

In order to use PrefixSpan we have to decide the ordering of input sequences. Currently we simply sort destination IPs in ascending order and store these sequences for each source IP. This input is then processed with the PrefixSpan algorithm. PrefixSpan supports several options for the processing process such as *minimal support* — minimal number of occurrences of sequences and *minimal length of a sequence*. The output is in form of frequent sequences and number of their occurrences. Experiments done so far produced many sequences (sometimes very long) with more than three occurrences (this naturally depends on the set of restrictions applied on the input data). Processing these destinations separately reveals corresponding source IPs that performed these communications. This approach can significantly help to decrease the matrix dimensions and to concentrate on a very specific subset(s) of the original matrix.

4.4 Behavioural vectors and user profiling

After applying some of the restrictions described in Section 4.3, we obtain a matrix of behavioural vectors. The rows of the matrix describe each source IP behaviour and form a vector. These vectors are ready to be processed using cluster analysis (or some other technique that calculates similarities) to find sets of source IPs 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 results. The questions we have to answer are what form of similarity measure we would like to have 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 the cosine similarity measure which is very widely used in data mining and works with vectors. Cosine similarity measures the angle between two vectors, so it perfectly fulfills our second requirement — how to express the "degree" of similarity. The formula for cosine similarity between two vectors of same dimensions $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots, b_n)$ is:

$$cosim(\varphi) = \frac{A \cdot B}{|A||B|}, \tag{32}$$

where $A \cdot B$ is

$$A \cdot B = \sum_{i=1}^{n} a_i b_i, \tag{33}$$

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

$$|A| = \sqrt{\sum_{i=1}^{n} a_i^2}. \tag{34}$$

So, finally

$$cosim(\varphi) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\sqrt{\sum_{i=1}^{n} b_i^2}}. \tag{35}$$

The output of *cosim* ranges from 0 (vectors are completely unrelated) 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 1 and 2). Given some information about destination IPs (Section 4.3.2) 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 destination so we tried to utilize this information. Given a vector of behaviour it is obvious that some components are more relevant for a given set of source IPs while some of them are less (for example Google is very popular and almost everyone visits this destination). IDF index ranges from 1 — this is the case when a given destination is not very relevant with respect to the set of sources — and $\log_2(\frac{1}{n})$ — this is the case when a given destination is highly relevant with respect to the given set of sources (only one source actually visited that destination). If we calculate IDF values for all destinations we will obtain a vector whose size is the same as for behavioural vectors. Each component of IDF vector represents actual relevancy of corresponding destination IPs. We use this IDF vector to balance behavioural vectors with relevances. This helps to emphasize components in behavioural vectors that are of greater relevance. Using this approach, vectors that are highly similar will be put even closer (compared to *cosim* metric without IDF) and vectors whose similarity is low will be put even more far away (compared to *cosim* metric 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.

The technical aspects of this approach are quite straightforward. After we have applied some restrictions and built a matrix of vectors we store this into a database table (we may call this the training set). Along with this, we calculate IDF values for all destination IPs and put them into database. Then we apply the same set of restrictions but take a different period of time (e.g., month) and again store vectors into a database table (we may call this the testing set). The calculation itself is done in a way that it is firstly computed using just the *cosim* metric and secondly using *cosim* influenced by the IDF

A	:	1(A, 1), 1(B, 1), 1(C, 1), 1(O, 1), 0.164399(E, 1)
B	:	1(A, 1), 1(B, 1), 1(C, 1), 1(O, 1), 0.164399(E, 1)
C	:	1(A, 1), 1(B, 1), 1(C, 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(C, 1), 0.124035(O, 1)
F	:	1(F, 1)
G	:	1(G, 1)
H	:	1(L, 1)
I	:	1(I, 1), 1(N, 1)
J	:	1(J, 1), 0.0905358(D, 1)
K	:	1(K, 1)

(a) Only *cosim* values, without IDF

A	:	1(A, 1), 1(B, 1), 1(C, 1), 1(O, 1), 0.0763637(E, 1)
B	:	1(A, 1), 1(B, 1), 1(C, 1), 1(O, 1), 0.0763637(E, 1)
C	:	1(A, 1), 1(B, 1), 1(C, 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(C, 1), 0.0573459(O, 1)
F	:	1(F, 1)
G	:	1(G, 1)
H	:	1(L, 1)
I	:	1(I, 1), 1(N, 1)
J	:	1(J, 1), 0.0661965(D, 1)
K	:	1(K, 1)

(b) With IDF values

Table 1: Similarity values

vector. For each source IP address from the training set we calculate similarities with all source IPs from the testing set. The results (i.e., similarity value with/without IDF, number of common destination IPs) are stored in a database table.

There are four tables with some results from the similarity measurement process (Tables 1 and 2). The first subtable (1a) corresponds to *cosim* values only (without the influence of IDF values) while the second one (1b) reflects IDF to optimize behavioural vectors. After a capital letter (a source IP address from the training data set) there is a list of most likely similar source IPs 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 IPs.

	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
C	0	0	0	0	0	0	0	0	279
D	0	0	37	0	0	0	1	0	0
E	0	0	0	0	32	0	0	0	4
F	0	0	0	24	0	0	0	0	0
G	23	0	0	0	0	0	0	0	0
H	0	0	0	0	0	22	0	0	0
I	0	0	0	0	0	0	0	19	0
J	0	0	0	0	0	0	17	0	0
K	0	15	0	0	0	0	0	0	0

(a) Training sets

	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
C	0	0	0	0	0	0	0	0	218
G	26	0	0	0	0	0	0	0	0
E	0	0	0	0	12	0	0	0	2
K	0	13	0	0	0	0	0	0	0
J	0	0	0	0	0	0	12	0	0
D	0	0	11	0	0	0	1	0	0
I	0	0	0	0	0	0	0	9	0
F	0	0	0	7	0	0	0	0	0
L	0	0	0	0	0	5	0	0	0
M	0	0	5	0	0	0	0	0	0
N	0	0	0	0	0	0	0	3	0
O	0	0	0	0	0	0	0	0	3

(b) Testing sets

Table 2: Behavioural vectors

So for example for source address D, the list of most similar IP addresses contains M and D (cf. Table 1). We can see that the similarity values are quite close to each other. Numbers in brackets indicate the count of common destinations with address D. So there is only one common destination for D and M and two common destinations for D and D. The behavioural vectors of all involved source addresses can be found in Tables 2a and 2b. We can see that in the testing set, source address D performed one communication with destination 7 and that is what caused that little difference in the similarities.

The subtables in Table 2 contain behavioural vectors for all source (capital letters) and destination (numbers) IPs involved in the similarity measurement process. Numbers in tables indicate the amount of communication that the set of source IPs did. The first

Subtable (2a) may be referred as to a "training data set" while the second one (2b) is used as a "testing data set". In accordance to the agreement with our IT department all real source and destination IPs were covered up.

This approach needs a further analysis and evaluation, but so far it seems to be quite effective in similarity computation and produces valuable results. We also plan to evaluate other similarity measures in the future for comparison.

4.5 Conclusion and future work

In this section we have presented current work on user profiling, particularly the data preparation phase, which is crucial for the profiling to be both significant and successful. We discussed the data that we work with and since the input is extremely large, we pointed out that methods for its optimization are of a great importance. It would not be reasonable to work with the whole dataset, because it consists of several interrelated parts and putting them onto the same level would decrease the significance of results. We discussed our methods for decreasing the matrix of behavioural vectors so as to make it contain relevant information; its dimensions are of some reasonable size; and the number of zero cells is lowered. We use histogram clustering to identify different levels of activity and entropy, together with IDF, to identify different levels of destinations. This approach seems to work well and helps a lot to optimize the input data in terms of relevance. During the process of "level identification", some information is dropped, but once we have the levels formed we can use more input data as a basis for a more detailed investigation.

Evaluations done so far showed that these methods of data preparation have a positive impact on the final clustering and increase the significance of results. At this time, applying these restriction rules is only "semi-automatic", but we intend to try to make the preprocessing phase more automatic in the future. This automatization will help the model to be able to process larger amounts of various input data in batches. This will help to evaluate the model under different initial conditions.

In the following sections we described the method we use to calculate similarities. We use cosine similarity with the extension of IDF relevancy values. This combination seems to add more accuracy as if it would be in the case of cosine similarity as such. We provided and discussed some results and illustrated that this approach looks very promising.

Since there is (still) no way of getting the name of a real person who used a source IP, we need to split the data into training and testing parts. The latter will be pseudonymized, which will let us verify whether our predictions based on the training set were true or false.

5 Influence of misinformation on anonymity metrics

5.1 Introduction

Information-theoretic metrics as presented in [KM07, 5. Metrics] can be used to measure the uncertainty of an adversary about identities. The better an adversary can assign a message or an action to an identity the smaller is the uncertainty and vice versa. The greater the adversary's uncertainty, however, the better is anonymity protected. Anonymity is best preserved, if identities cannot be distinguished by means of identifying one which is more (or less) likely related to the message or action. That is, guessing is the best the adversary can do. No anonymity remains, if the adversary is certain that the message belongs to a particular identity. These two cases and the shade between them can be represented as probability distribution. That is, the adversary's knowledge is modelled as a probability for each identity with each probability rating the case that the respective identity is related to the message. The overall uncertainty of the adversary expressed in these probabilities is then exactly the entropy of the probability distribution. Therefore, entropy is used as a metric for anonymity, either as degree of anonymity, that is set in relation to the maximum entropy, or as the amount of information which the adversary still lacks in order to identify anyone.

In this section, we point out what effect misinformation has on information-theoretic privacy metrics. That is, we elaborate on how and to which degree these metrics are affected, if the probability distribution does not only base on true information. In fact, the incorporation of misinformation can only happen, if the adversary is not able to verify all data. This might be likely for arbitrary internet scenarios, but is very unlikely to happen with significant impact in census scenarios, for instance. However, we will not go into detail about the costs of generating misinformation or verifying data.

The results are valid for the metrics proposed in [KM07, 5.2.1 Risk of Re-identification], [KM07, 5.3 Data-flow in Networks], [KM07, 5.4 Generalizations]. However, these approaches differ in the flavour of entropy they utilize, cf. [KM07, 5.4.4 Rényi Entropy]. Therefore, misinformation will be discussed with respect to Shannon entropy in Section 5.2 and with respect to Rényi entropy in Sections 5.3 and 5.4.

5.2 Average-case anonymity

In this section, we assume that the measurement result is known, i.e., an entropy value $H_{\varnothing}$, as well as the number of elements n of the source and if possible also the probability distribution of the source elements, where $H_{\varnothing}$ is calculated from. Further, we assume an error e, which denotes the maximum displacement of the probabilities of the source elements.

In order to derive the maximum possible deviation of the measurement result we first need to find out under which conditions a constant change of probabilities affects Shannon entropy most. This can either happen if the change leads to greater entropy or if it

leads to less entropy. That is, the more an error e corresponds to a given probability distribution (i.e., strengthens the element with the highest probability) the more Shannon entropy is decreased. Accordingly, the more the error e leads to uniform distribution of probabilities the more Shannon entropy is increased.

Increasing entropy. The greatest increase in entropy can be achieved by increasing the probability of the least probable element[5]. For showing that, we assume n identities, where $n-1$ identities have equal probability. While increasing the probability p_n of the remaining identity from zero to $\frac{1}{n}$, Shannon entropy reduces to the sum of the uncertainty given by p_n, that is $p_n \log_2(p_n)$, and the uncertainty given by all other probabilities in the distribution. We know that all other probabilities are uniformly distributed. Thus, the sum of all other probabilities is determined by the remaining probability $1-p_n$ and its $n-1$ equal shares. That is, we know that each share is exactly $\frac{1-p_n}{n-1}$. In sum, this can be simplified to Equation (36):

$$H_{\varnothing}(p_n) = -p_n \log_2(p_n) - (1-p_n) \log_2 \frac{1-p_n}{n-1} \tag{36}$$

Suppose, for instance, n identities with $n = 10$ whereof nine have equal probabilities. Figure 7 shows the Shannon entropy for $0 < p_n \leq \frac{1}{10}$. We see that a constant change of p_n increases the entropy most, if the probabilities are uniformly distributed, that is $p_n = \frac{1}{10}$. The higher the value of p_n was before, the less does the change affect the entropy.

Decreasing entropy. Accordingly, the opposite change of probabilities has the greatest effect. That is, the effect can be achieved by decreasing the probabilities which previously had the lowest probability within the probability distribution while increasing the probability of the element with the highest previous probability. This leads to the most heterogeneous distribution caused by a constant change to probabilities, and by that to the smallest possible entropy.

We see that anonymity metrics which base on Shannon entropy can be confused by misinformation. The greater the entropy becomes with incorporated misinformation the more the metric indicates a too strong anonymity compared to the reality and vice versa. The upper bound of entropy is approached, if the probabilities of the identities with lowest probabilities are increased while the probabilities of the identities with the highest probabilities are decreased. Similarly, the lower bound is approached by decreasing probabilities of as many elements as possible to zero while the probability of the element with the highest original probability is increased.

[5] or plural, that is increasing the probability of the least probable elements in case there is more then one element with the smallest probability value

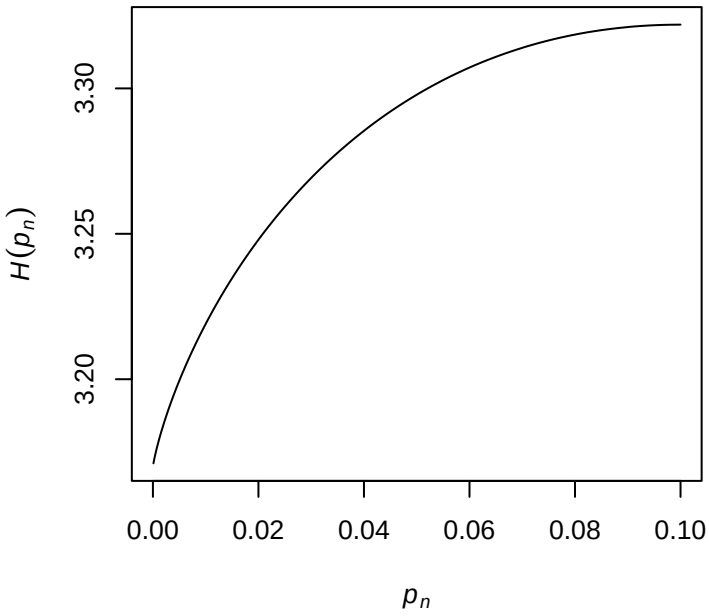

Figure 7: Shannon entropy for $0 < p_n \leq \frac{1}{10}$

5.3 Individual anonymity

The anonymity of a single identity in a system is determined by the probability that the adversary assigns to the identity. We call this individual anonymity. The metric of individual anonymity, as described in [KM07, 5.4.4.1 Quantifying Anonymity, Equation 5.25], is the amount of information which an adversary can derive from the identity in a given probability distribution. The maximum effect of an error e on the measurement depends only on the probability of the identity. In the following, we denote the identity in question by i. For a given probability $p(i)$ the upper bound for the individual anonymity can be determined as $H_{max,e}(i)$.

$$H_{\max,e}(i) = \begin{cases} -\log_2\big(p(i) - e\big) & \text{iff } p(i) - e > 0 \\ \text{undefined} & \text{otherwise} \end{cases} \tag{37}$$

Accordingly, the lower bound can be determined as $H_{min,e}(i)$.

$$H_{\min,e}(i) = \begin{cases} -\log_2(p(i) + e) & \text{iff } p(i) + e < 1 \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

5.4 Worst-case anonymity

The worst-case anonymity is the amount of information that the adversary lacks in order to identify $i_{\max}$ where $i_{\max}$ denotes the identity with highest probability within

a given distribution. Worst-case anonymity is therefore a special case of individual anonymity. The metric for worst-case anonymity is specified in [KM07, 5.4.4.1 Quantifying Anonymity, Equation 5.26].

The lower bound of worst-case anonymity is $H_{\min,e}(i_{\max})$. The upper bound depends on the shape of the given probability distribution. The probabilities of the elements with the highest probabilities need to be decreased by the amount e in order to reach the lowest possible value for the element with the highest probability. $H_{\max,e}(i_{\max})$ determines the upper bound.

5.5 Conclusion

We have shown that misinformation can falsify information-theoretic privacy metrics such that they either indicate stronger privacy than there is in reality or less. We have described the upper and lower bounds of a possible falsification by misinformation. Both, a possible adversary and a user may have an interest in emitting misinformation: The case of a metric indicating stronger anonymity on the adversary's side can be seen as confusion of the adversary. This might be a goal of the user. On the user's side, however, it could lead to the false sensation of too much remaining anonymity. This could be a goal of the adversary, accordingly. The case of a metric indicating lower anonymity on the adversary's side leads to a faster de-anonymization of identities. However the adversary does not necessarily learn the true identity, since his conclusion is still based on misinformation. On the user's side, the indication of lower anonymity could lead to frustration and thus prevent the user from performing harmless actions, that is, actions that would not reveal much about the identity in reality. This case is probably not of interest for anyone.

6 Conclusion and outlook

In the series of three deliverables within Workpackage 13 *Privacy Fundamentals*, namely D13.6 *Privacy Modelling and Identity*, this one D13.8, and the forthcoming D13.9 *Estimating quality of identity*, the present deliverable is a step further on the way towards the concepts of measurement of anonymity which will in the third deliverable then be used and applied to estimating the quality of identity using real-world examples.

Several aspects of approaches presented in Deliverable D13.6 appear to be problematic, for examples some conditions in the context of unobservability show to be neither necessary nor fulfilled in the case of the formal transcription of common criteria. We have shown some of the shortcomings, others might appear. Clearly this is not intended to put the concepts down, but meant to raise awareness of the sensitivity of possible measures within the field of identity and privacy — sensitivity with respect to the goal, the environment, the attacker, the situation, etc. There seems to be no "general" tool for measuring in this context, some might be applicable in some situations, but some have general problems as presented in this deliverables. Further work must be done in this area to identify other problematic issues.

The general problem of combining information has been discussed with respect to a Bayesian approach, as well as its application to the computation of anonymity in the case where an adversary has access to profile information and anonymous network traffic observations. Combination using a Bayesian method means also that its result is a probability distribution of the senders (or receivers) of messages. This probability distribution is then again input for the metric which computes sender (and receiver) anonymity by means of entropy. Further investigations within this context might consider the applicability and use of different approaches of combining information from different sources and especially also integrate the notion of reliability of the sources. In this sense, the general assumption of a given prior probability distribution is often surely true, but sensitivity analysis, i.e., determining to which extend small changes in the prior probabilities (and especially in which of them) will affect the results, might reveal results with respect to the dependence of the correctness of such a distribution. This points also into the direction of considering the influence of misinformation on anonymity metrics as has been done in Section 5.

Using a larger example in the context of IP connections, we have shown problems occurring in the case of profiling where lots and lots of data are available, i.e., typically too big to start profiling algorithms from scratch. We have provided an introduction to the current work on user profiling, particularly on the data preparation phase. This phase is clearly crucial for the profiling to be both significant and successful, especially in the case of such very large data sets. Methods for the optimization of such large data sets are of a great importance as it would not be reasonable to work with the whole dataset, because it consists of several interrelated parts and putting them onto the same level would decrease the significance of results. The method discussed focuses on decreasing the matrix of behavioural vectors so as to make it contain relevant information and

"compactifies" the matrix. A key element is histogram clustering to identify different levels of activity and entropy, and — together with inverse document frequency IDF — to identify different levels of destinations. Evaluations show that this technique works well, i.e., the data preparation phase has positive impact on the clustering and increases significance. Note that using this technique it is also possible to (re-)integrate more data after the process of "level identification", where parts of the data has been dropped during computation. At this time, applying these restriction rules is only "semi-automatic"; future work is done in the direction of getting this more automatic.

We showed then how so-called cosine similarities can be used to compute similarities of the resulting behavioural vectors. These similarity values are then compared with an extension thereof which includes inverse document frequency values. Some results show that the latter appears to be more accurate. As in the presented example of IP connections, no "real testing data" is available and in order to have a possibility to even have some test results, we need to split the available data into training and testing parts. The testing data is pseudonymized, which allows to measure the quality of the predictions based on the training set.

Misinformation has typically impacts on information-theoretic privacy metric and is therefore on the one hand a problem when the user is not aware of its presence (or absence), but on the other hand might also be some sort of a tool which can be used for different purposes, so for example in order to confuse the adversary or by giving the user a false impression of being more anonymous than actually is the case. We have shown how changes in the probability distribution influence the privacy metrics, and hence implications in different directions might arise from such (mis-)information.

The forthcoming Deliverable D13.9 will focus on estimating quality of identities that will extend work by showing (if possible) how theoretical models may be used for real-world scenarios. The result should describe the ways to estimate quality of identities in some real-case scenarios, with the vision to involve some distinct technologies identified in other work of FIDIS, namely of WP3.

References

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

[Cha81] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[Cla06] S. Clauß. A framework for quantification of linkability within a privacy-enhancing identity management system. In *Emerging Trends in Information and Communication Security*, volume 3995 of *Lecture Notes in Computer Science*, pages 191–205. Springer, 2006.

[CM07] D. Cvrček and V. Matyáš, editors. *D13.1: Identity and impact of privacy enhancing technologie, Deliverable of FIDIS' Workpackage 13*. FIDIS `http://www.fidis.net`, 2007.

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

[CS06] S. Clauß and S. Schiffner. Structuring anonymity metrics. *Proceedings of the ACM Workshop on Digital Identity Management*, pages 55–62, 2006.

[Dan04] G. Danezis. The traffic analysis of continuous-time mixes. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, volume 3424 of *Lecture Notes in Computer Science*, pages 35–50. Springer, May 2004.

[DCSP02] C. Diaz, J. Claessens, S. Seys, and B. Preneel. Information theory and anonymity. In B. Macq and J.-J. Quisquater, editors, *Werkgemeenschap voor Informatie en Communicatietheorie*, pages 179–186, 2002.

[DDM03] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type III anonymous remailer protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pages 2–15, 2003.

[DMS04] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, pages 303–320. USENIX, 2004.

[DP04] C. Diaz and B. Preneel. Reasoning about the anonymity provided by pool mixes that generate dummy traffic. In *Information Hiding*, volume 3200 of *Lecture Notes in Computer Science*, pages 309–325. Springer, 2004.

[DS03] C. Díaz and A. Serjantov. Generalising mixes. In R. Dingledine, editor, *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*, volume 2760 of *Lecture Notes in Computer Science*, pages 18–31. Springer, March 2003.

[DS04] G. Danezis and A. Serjantov. Statistical disclosure or intersection attacks on anonymity systems. In *Proceedings of 6th Information Hiding Workshop (IH 2004)*, volume 3200 of *Lecture Notes in Computer Science*, Toronto, May 2004. Springer.

[DSCP02] C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *Designing Privacy Enhancing Technologies, Proceedings of PET'02*, volume 2482 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 2002.

[DTD07] C. Diaz, C. Troncoso, and G. Danezis. Does additional information always reduce anonymity? In T. Yu, editor, *Workshop on Privacy in the Electronic Society 2007*, pages 72–75. ACM, 2007.

[KEB98] D. Kesdogan, J. Egner, and R. Büschkes. Stop-and-go MIXes: Providing probabilistic anonymity in an open system. In *Proceedings of Information Hiding Workshop (IH 1998)*, volume 1525 of *Lecture Notes in Computer Science*, pages 83–98. Springer, 1998.

[KM07] M. Kumpost and V. Matyáš, editors. *D13.6 Privacy Modelling and Identity, Deliverable of FIDIS' Workpackage 13.* FIDIS `http://www.fidis.net`, 2007.

[LB04] J.-W. Lee and D.-K. Baik. A Model for Extracting Keywords of Document Using Term Frequency and Distribution. In A. 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.

[MD04] N. Mathewson and R. Dingledine. Practical traffic analysis: Extending and resisting statistical disclosure. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, volume 3424 of *Lecture Notes in Computer Science*, pages 17–34. Springer, May 2004.

[PHA+01] J. Pei, J. Han, M. B. Asl, H. Pinto, Q. Chen, U. Dayal, and M. C. Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In *ICDE '01: Proceedings of the 17th International Conference on Data Engineering*, page 215, Washington, DC, USA, 2001. IEEE Computer Society.

[R D06] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2006. ISBN 3-900051-07-0.

[SD02] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Designing Privacy Enhancing Technologies, Proceedings of PET'02*, volume 2482 of *Lecture Notes in Computer Science*, pages 41–53. Springer, 2002.

[SDS02] A. Serjantov, R. Dingledine, and P. Syverson. From a trickle to a flood: Active attacks on several mix types. In F. Petitcolas, editor, *Proceedings of Information Hiding Workshop (IH 2002)*, volume 2578 of *Lecture Notes in Computer Science*. Springer, October 2002.

[Sha48] C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423:623–656, 1948.

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

[THV04] G. Tóth, Z. Hornák, and F. Vajda. Measuring anonymity revisited. `http://www.freehaven.net/anonbib/cache/THV04.pdf` (12.3.2008), 2004.

[War63] J. H. Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.