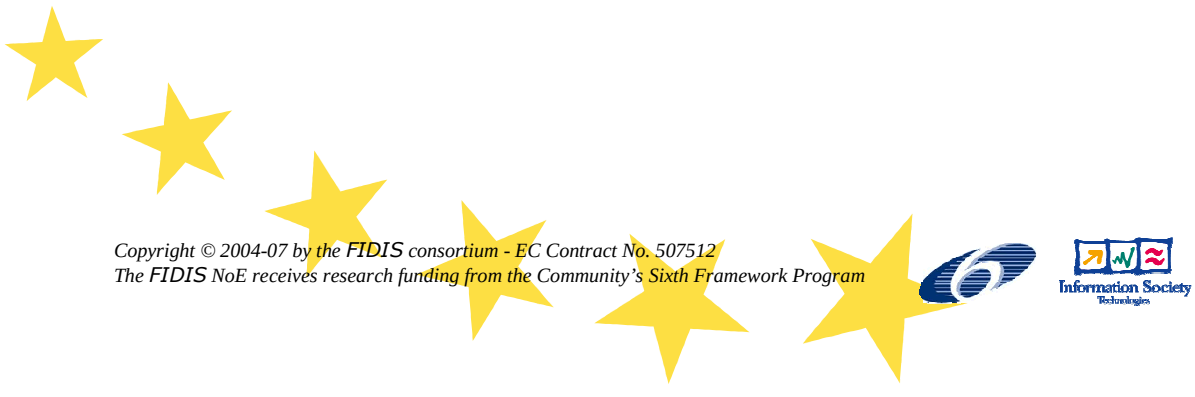
FIDIS

Future of Identity in the Information Society

Title:	"D13.6: Privacy modelling and identity"
Author:	WP13
Editors:	Marek Kumpošt (MU, Czech Republic) Vashek Matyáš (MU, Czech Republic) Stefan Berthold (TU Dresden, Germany)
Reviewers:	Hans Buitelaar (Tilburg U., Netherlands) Claudia Diaz (COSIC, KU Leuven)
Identifier:	D13.6
Type:	[Deliverable-Report]
Version:	1.0
Date:	Thursday, 22 November 2007
Status:	[Final]
Class:	[Public]
File:	fidis-wp13-del13.6.final.pdf

Brief Summary

This document critically reviews existing approaches (most common theoretical tools) for modelling relations of identity related information and also some related aspects of their applicability for measurement or quantitative expression of (the level of) privacy.



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	22.07.2007	• Initial draft
0.3	04.09.2007	• Updated version of chapters • Major formatting updates
0.7	16.10.2007	• Updated version of chapters • Minor formatting updates
0.8	23.10.2007	• Submitted for internal review
0.9	19.5.2007	• Revision after internal review
1.0	22.11.2007	• 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:

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

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

Management Summary

This report aims to provide a comprehensive insight into privacy (and context) modelling approaches, namely into technical and formal approaches to privacy quantification. Majority of models available rely on *context information* – information that describes entity (typically user) behaviour while interacting with a system. Context information can be seen, e.g., in the form of log files containing information about all HTTP communications with a web server. Another example can be a log file containing search queries entered into a search engine. While this type of information is valuable for some profiling techniques that enable the system to serve customized content, it also forms a solid basis for retrieving sensitive information about individuals. Therefore special care has to be taken while analyzing and processing this data.

Privacy/content models discussed in this deliverable aim to process the data to learn frequent behavioral patterns as well as to decide how sensitive information this data may contain. Metrics for privacy quantification then aim to measure the degree of privacy a protocol or a communication system can provide to its users. Another purpose of such quantification is to provide a metric for the (level of) privacy which users actually may expect with respect to their situation, e.g., considering their previous actions.

Deliverable D13.1 and this deliverable 13.6 are now essential for the follow-up work of FIDIS WP13, where the planned deliverables are as follows:

- *Deliverable D13.8: Applicability of privacy models*, where we plan to use some privacy modelling approaches in use-cases involving profiling, systems using different forms of identities, etc. The goal of that deliverable will be to review/illustrate the applicability of models from this deliverable D13.6, and this deliverable actually went few steps ahead in this way.
- *Deliverable D13.9: Estimating quality of identities* that will extend our previous 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.

Contents

1 Introduction **3**

2 Concepts and Terminology **5**
2.1 Privacy and Its Threats 5
2.2 Pfitzmann-Hansen Terminology 6
2.2.1 Anonymity 7
2.2.2 Unlinkablity 7
2.2.3 Unobservability 8
2.2.4 Pseudonymity 9
2.3 k-Anonymity Model 9
2.4 Common Criteria 10
2.4.1 A simple example 12
2.4.2 Privacy in the Common Criteria 13

3 Context information **14**
3.1 Context information 14
3.1.1 Categorization of context information 15
3.1.2 Quality of context information (QoCI) 16
3.2 (Un)linkability 17
3.3 Common Criteria revisited 17
3.3.1 Unlinkability – The Unlinkables 19

4 Context information, user behaviour and privacy models **22**
4.1 Context information models 22
4.1.1 Freiburg Privacy Diamond 22
4.1.2 Categorization and modelling of quality in context information 27
4.1.3 Set theory 28
4.1.4 Directed graph 28

4.1.5 First-order logic . . . 29
4.1.6 modelling context information with ORM . . . 29
4.2 User behaviour models . . . 30
4.2.1 Dynamic vs. static behavioral models . . . 30
4.2.2 Global mixture model . . . 30
4.2.3 Maximum entropy model . . . 31
4.2.4 Hidden Markov Model . . . 31
4.3 Privacy models . . . 32
4.4 Data mining . . . 32
4.5 Cluster analysis . . . 33
4.5.1 Measuring distance between objects . . . 33
4.5.2 Hierarchical clustering – measuring distance between clusters . . . 35
4.6 Different types of similarity measures . . . 35
4.6.1 Usage based measure . . . 36
4.6.2 Frequency based measure . . . 36
4.6.3 Viewing-time based measure . . . 36
4.6.4 Visiting-order based measure . . . 37

5 Metrics **38**
5.1 Formal Methods . . . 39
5.1.1 Function Views . . . 40
5.1.2 Formal Languages and Semantics . . . 41
5.2 Persistent Data & Statistical Databases . . . 42
5.2.1 Risk of Re-identification . . . 43
5.2.2 k-Anonymity . . . 44
5.3 Data-flow in Networks . . . 44
5.4 Generalizations . . . 47
5.4.1 Local Anonymity . . . 47
5.4.2 Towards Arbitrary Attributes . . . 47
5.4.3 Unlinkability . . . 54
5.4.4 Rényi Entropy . . . 56

6 Conclusions **62**

Bibliography **71**

Chapter 1

Introduction

The aim of this deliverable is to provide a comprehensive insight into privacy (and context) modelling approaches, namely into technical and formal approaches to privacy quantification. Majority of models available rely on *context* information – information that describes entity (typically user) behaviour while interacting with a system. Context information can be seen, e.g., in the form of log files containing information about all HTTP communications with a web server. Another example can be a log file containing search queries entered into a search engine. On one hand, this type of information is valuable for some profiling techniques that enable the system to serve customized content. On the other hand, context information can be considered as a solid basis for retrieving sensitive information about individuals, as is witnessed in FIDIS deliverables related to profiling. Therefore special care has to be taken while analyzing and processing this data.

Privacy/content models discussed in this deliverable aim to process the data to learn frequent behavioral patterns as well as to decide how sensitive information this data may contain. Metrics for privacy quantification then aim to measure the degree of privacy a protocol or a communication system can provide to its users. Another purpose of such quantification is to provide a metric for the (level of) privacy which users actually may expect with respect to their situation, e.g., considering their previous actions.

In the second chapter we start with the description of some basic concepts and terminology used in the field of privacy modelling and privacy quantification. We discuss privacy as a complex concept with different meaning to

different people and point out currently known threats. Four main privacy properties – anonymity, pseudonymity, unlinkability and unobservability are described according to Pfitzmann and Hansen definitions, and then discussed from the Common criteria point of view.

Chapter 3 is dedicated to context information, its categorization and quality. Context information is a critical source of knowledge for both the privacy modelling and privacy quantification. The rest of this chapter deals with a revisited view of Common criteria privacy properties, where the proposed definitions allow to quantify all privacy properties in a probabilistic way.

Chapter 4 is organized as a survey of selected context information and user behaviour modelling approaches. We start with context information models – Freiburg Privacy Diamond, PATS (which is based on probabilistic definitions of four privacy properties), models based on set theory, directed graphs, first-order logic and finally a variant of Object-Role modelling. Second part of this chapter deals with behaviour modelling techniques. We discuss global mixture model, maximum entropy model and hidden Markov model. Data mining (specifically the cluster analysis) and its importance (and the way it is used) in user behaviour modelling is shortly introduced at the end of the chapter 4.

Chapter 5 presents existing metrics for privacy properties quantification (anonymity, pseudonymity, unlinkability, unobservability). It starts with formal methods that have been proposed for determination of the degree of anonymity, namely "function views". The following part addresses the question of privacy in (statistical) databases with respect to possible risk of entities' re-identification and the degree of user's anonymity provided in systems like the DC net, mix-based systems or Crowds. Section 5.4 outlines generalizations of the previous approaches and refers to the work which is currently state of the art.

Chapter 2

Concepts and Terminology

2.1 Privacy and Its Threats

Privacy is a complex and subjective concept with different meanings to different people, that depend on the context in which it is used. Solove presented in [Sol06] a taxonomy of privacy from the perspective of law, where 16 different types of privacy violation are defined. The author classifies the identified privacy violations in four categories, which are:

Information Collection surveillance and interrogation.

Information Processing aggregation, identification, insecurity, secondary use and exclusion.

Information dissemination breach of confidentiality, disclosure, exposure, increased accessibility, blackmail,appropriation and distortion.

Invasion intrusion and decisional interference.

Technological measures to protect against privacy violations focus mostly on preventing the unintended leakage of information; while other types of violations fall out of the scope of technological systems and legal measures are needed in order to prevent them. Technical systems can better protect against the following particular privacy threats:

Surveillance considering adversary capable of monitoring electronic transactions, privacy-enhancing technologies aim to reduce the risk of surveillance by concealing information about the content and circumstances of electronic transactions from adversary. When users are able to keep transaction contents *confidential* and to act *anonymously*, they protect themselves against surveillance threats.

Interrogation the technical property the protects a user from being forced to disclose information is called *plausible deniability*. Systems that provide plausible deniability make it impossible for adversary to prove that the user is concealing information.

Aggregation the property that prevents the aggregation of information as related to each other or to a particular subject is *unlinkability*.

Identification Identification is connecting data to individuals. Anonymity, unlinkability and confidentiality properties prevent this connection to be revealed.

In order to preserve privacy in electronic applications, information must be made unavailable to potential adversaries trying to identify, profile, or link subjects with actions, attributes or other subjects. In the remaining of this section, we present definitions of the main privacy properties that have been subject of research, such as anonymity, unlinkability, unobservability, and pseudonymity.

The quantification of these properties is achieved by the use of metrics. Metrics allow for the comparison and evaluation of the level of privacy provided by different systems. In Chapter 5, we present the existing metrics for quantification of privacy properties.

2.2 Pfitzmann-Hansen Terminology

Pfitzmann and Hansen [PH01] proposed in 2000 a set of working definitions for anonymity, unlinkability, unobservability, and pseudonymity. These definitions have since been adopted in most of the anonymity literature. Their authors continue releasing regular updates on the document addressing feedback from the research community. The latest versions of the document are

publicly available at `http://dud.inf.tu-dresden.de/Anon_Terminology.shtml`.

2.2.1 Anonymity

There always has to be an appropriate set of subjects with potentially the same attributes to enable anonymity of a subject. Anonymity is thus defined as *the state of being notidentifiable within a set of subjects, the anonymity set.*

The *anonymity set* is a set of all possible subjects. With respect to acting entities, the anonymity set consists of the subjects who might cause an action. With respect to addressees, the anonymity set consists of the subjects who might be addressed. Both anonymity sets may be disjoint, be the same, or they may overlap. The anonymity sets may vary over time.

According to the Pfitzmann-Hansen definition of anonymity, the subjects who may be related to an anonymous transaction constitute the *anonymity set* for that particular transaction. A subject carries on the transaction *anonymously* if he cannot be distinguished (by an adversary[1]) from other subjects. This definition of anonymity captures the probabilistic information often obtained by adversaries trying to identify anonymous subjects.

2.2.2 Unlinkablity

The ISO15408:1999 defines unlinkability as follows:

"[Unlinkability] ensures that a user may make multiple uses of resources or services without others being able to link these uses together. [...] Unlinkability requires that users and/or subjects are unable to determine whether the same user caused certain specific operations in the system."

This notion is restricted to users, while it makes sense to generalize it to arbitrary items within a given system (e.g., between users and services used or between different uses of services).

[1] We shall use the terms of *adversary* and *attacker* as synonymous through this document.

Further we may differentiate between *absolute unlinkability* (as in the given definition; i.e., "no determination of a link between uses") and *relative unlinkability* (i.e., "no change of knowledge about a link between uses"), where *relative unlinkability* between arbitrary items could be defined as follows:

Unlinkability of two or more Items Of Interest (IOIs; e.g., subjects, messages, events, actions, ...) means that within the system (comprising these and possibly other items), from the attacker's perspective, these items of interest are no more and no less related after his observation than they are related concerning his a-priori knowledge.

This means that the probability of those items being related from the attacker's perspective stays the same before (a-priori knowledge) and after the attacker's observation (a-posteriori knowledge of the attacker). Roughly speaking, unlinkability of items means that the ability of the attacker to relate these items does not increase by observing the system.

2.2.3 Unobservability

In contrast to anonymity and unlinkability, where not the IOI, but only its relationship to IDs or other IOIs is protected, for unobservability, the IOIs are protected as such. *Unobservability is the state of items of interest (IOIs) being indistinguishable from any IOI (of the same type) at all.*

This means that messages are not distinguishable from *random noise*. As we had anonymity sets of subjects with respect to anonymity, we have unobservability sets of subjects with respect to unobservability. Sender unobservability then means that it is not noticeable whether any sender within the unobservability set sends a message. Recipient unobservability then means that it is not noticeable whether any recipient within the unobservability set receives a message. Relationship unobservability then means that it is not noticeable whether anything is sent out of a set of could-be senders to a set of could-be recipients. In other words, it is not noticeable whether, within the relationship unobservability set of all possible sender-recipient-pairs, a message is exchanged in any relationship.

2.2.4 Pseudonymity

Pseudonyms are identifiers of subjects (or sets of subjects when we generalize it a bit). The subject which the pseudonym refers to is the holder of the pseudonym.

Being pseudonymous is the state of using a pseudonym as ID

We assume that each pseudonym refers to exactly one holder, it is invariant over time, being not transferred to other subjects. Specific kinds of pseudonyms may extend this setting: A group pseudonym refers to a set of holders, i.e. it may refer to multiple holders; a transferable pseudonym can be transferred from one holder to another subject becoming its holder. Such a group pseudonym may induce an anonymity set: Using the information provided by the pseudonym only, an attacker cannot decide whether an action was performed by a specific person within the set.

Defining the process of preparing for the use of pseudonyms, e.g., by establishing certain rules how to identify holders of pseudonyms, leads to the more general notion of pseudonymity:

Pseudonymity is the use of pseudonyms as IDs

An advantage of pseudonymity technologies is that accountability for misbehaviour can be enforced. Also, persistent pseudonyms allow their owners to build a pseudonymous reputation over time.

2.3 k-Anonymity Model

A model for anonymizing personal records in a database has been proposed by Samarati and Sweeney in [SS98, Swe02a]. While anonymity at the communication layer needs to be protected from traffic analysis attacks, anonymized records may be vulnerable to re-identification. Re-identification is the process of relating unique and specific entities to seemingly anonymous data [Mal02], and as such, is an attack on the privacy of a data collection.

When a data holder wants to release anonymized personal records (e.g., for research purposes), it is not enough to remove obvious identifiers such as name, address, or national ID number. Often, some subset of the data

fields constitute a quasi-identifier. For example, ZIP code together with the gender and the birth date may be enough to re-identify a substantial number of *anonymized* data subjects.

A quasi-identifier is defined as: "Let $RT(A_1, ..., A_n)$ be a table and QI_{RT} be the quasi-identifier associated with it. RT is said to satisfy k-anonymity if and only if each sequence of values in $RT[QI_{RT}]$ appears with at least k occurrences in $RT[QI_{RT}]$."

The k-anonymity model assumes that there is some publicly available database (e.g., the census or voter registration list) that contains certain attributes for each of the data subjects included in it. When a second data set is released, it is often the case that, even if identifiers have been removed, quasi-identifiers can be found, such that re-identification (i.e., linking to the publicly available database in order to find the name, address, etc.) is possible.

k-anonymity is defined as follows [Swe02a, SS98]: "Let $RT(A_1, \ldots, A_n)$ be a table and QI_{RT} be the quasi-identifier associated with it. RT is said to satisfy k-anonymity if and only if each sequence of values in $RT[QI_{RT}]$ appears with at least k occurrences in $RT[QI_{RT}]$."

In other words, a set of records is k-anonymous if there are at least k records in the anonymity set for each possible quasi-identifier. The techniques proposed to make a set of data k-anonymous are based on suppression and generalization of data fields.

2.4 Common Criteria

Common Criteria [The99] is a standard used for security evaluations of IT products and systems. It defines, among many other issues, privacy as one of possible security properties of such systems. We first summarize the relevant Common Criteria notions and concepts, and then introduce the concepts of the privacy definition as defined in Common Criteria documents.

Target of Evaluation (TOE) – An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.

TOE Security Functions (TSF) – A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TOE security policy.

TSF Scope of Control (TSC) – The set of interactions that can occur with or within a TOE and are subject to the rules of the TOE security policy.

Subject – An entity within the TSC that causes operations to be performed.

Assets – Information or resources to be protected by the countermeasures of a TOE.

Object – An entity within the TSC that contains or receives information and upon which subjects perform operations.

User – Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

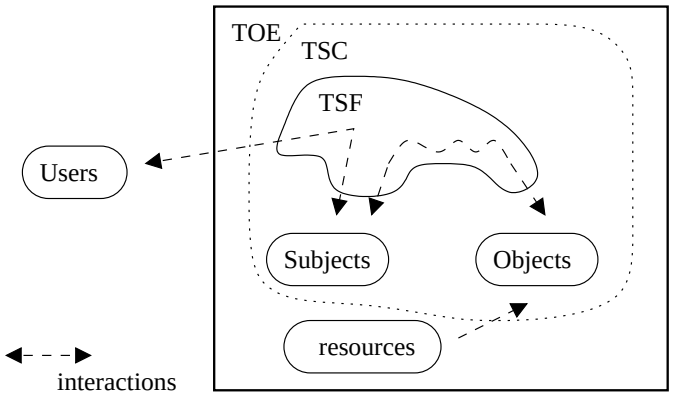


Figure 2.1: Common Criteria model.

We can see (fig. 2.1) that user does not access objects directly but through subjects – internal representation of herself inside TOE/TSC. This indirection is exploited later on for a definition of pseudonymity as we will see later. Objects represent not only information but also services mediating access to TOE's resources. This abstract model does not directly cover communication like in (remailer) mixes as it explicitly describes only relations

between users/subjects and resources of target information system. However, it is not difficult to extend the proposed formal definitions of major privacy concepts based on this model for communication models.

2.4.1 A simple example

Let us present a trivial example [MC04] that we use later in this chapter to compare the formal models for privacy. The attacker attempts to determine which payment cards are used by a certain person with a particular card – she is interested in linking together all the cards of this person (identification of the particular person is not part of the attacker's goal at the moment). We assume the attacker is able to collect payment receipts of shoppers from the same house or the same company. For this subset of supermarket clients we do not mind that a given receipt shows only a part of the payment card number.

There are three payment cards (with numbers 11, 21, 25) used for three actual shoppings (visits of the supermarket resulting in payments – A, B, C), and there is also a set of typical baskets/shopping lists (l, m) in our simplistic example.

The attacker has a precise (100%) knowledge about connections between payment cards and shoppings, and an imprecise knowledge about classification of individual shoppings into typical "consumer group" baskets. This classification to "typical baskets" is usually done with some kind of a data-mining algorithm over actual shopping lists. Note that one could obviously achieve perfect knowledge should loyalty cards be used (and their numbers on the receipts), introduction of this has no qualitative impact to this example illustration in our model.

With just changing semantics, we may define a very similar example based on users of chat services connecting from a given Internet cafe. The categories would then be chat-room pseudonyms, chat sessions, and classification into groups based on interest (content) and/or language, with the attacker's goal of identifying pseudonyms used by one user in different chat sessions.

2.4.2 Privacy in the Common Criteria

Unobservability – *This family ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used.* The protected asset in this case can be information about other users' communications, about access to and use of a certain resource or service, etc. Several countries, e.g. Germany, consider the assurance of communication unobservability as an essential part of the protection of constitutional rights. Threats of malicious observations (e.g., through Trojan Horses) and traffic analysis (by others than communicating parties) are best-known examples.

Anonymity – *This family ensures that a user may use a resource or service without disclosing the user identity. The requirements for Anonymity provide protection of the user identity. Anonymity is not intended to protect the subject identity.* Although it may be surprising to find a service of this nature in a Trusted Computing Environment, possible applications include enquiries of a confidential nature to public databases, etc. A protected asset is usually the identity of the requesting entity, but can also include information on the kind of requested operation (and/or information) and aspects such as time and mode of use. The relevant threats are: disclosure of identity or leakage of information leading to disclosure of identity – often described as "usage profiling".

Unlinkability – *This family ensures that a user may make multiple uses of resources or services without others being able to link these uses together.* The protected assets are of the same as in Anonymity. Relevant threats can also be classed as "usage profiling".

Pseudonymity – *This family ensures that a user may use a resource or service without disclosing its user identity, but can still be accountable for that use.* Possible applications are usage and charging for phone services without disclosing identity, "anonymous" use of an electronic payment, etc. In addition to the Anonymity services, Pseudonymity provides methods for authorisation without identification (at all or directly to the resource or service provider).

Chapter 3

Context information

We can learn a lot about an individual based on his *context* and in a best case can successfully predict possible future actions by combining several types of context information in an efficient way. This can reveal some *private* data about users and therefore privacy protection issues must be carefully taken into consideration. A critical term here is *user profile*. If we have a good profile of a user (and his behaviour) we can try to find and link together similar profiles and, with some probability, identify a user – in this case we do not know the real identification of a user, we just know that profiles in a certain set share some similarities.

3.1 Context information

In order to develop context-aware systems, we need a mechanism for effective context information processing and categorization. The aim is to develop a model that represents user's behaviour based on his past activity. Accuracy of the model is a key feature for predictions of future actions that are based on this model.

Context information is a type of information that is directly or indirectly connected to an individual or arises from his activity. Typical examples of user's activity [MC04] (considering IT world only) can be the time when user typically logs into a system, typed shell commands, visited IP addresses, number of ssh connections, types of services used on a network, size of e-mail

messages, search queries, length of sessions... By direct/indirect context information we mean e.g. information about friends and their interests (your social network), current/past geographic locations, some regular activities, health condition, etc.

Context information is associated with an individual and is a descriptive type of information about a particular entity. It is clear that context information is potentially sensitive and therefore it is necessary to have some privacy protection mechanisms for users. Knowledge of this information may have negative sociological, financial or material impact on you if it is abused by an attacker. Global tendency is therefore to develop efficient mechanisms for privacy protection (these are called Privacy Enhancing Technologies – PET). User privacy can be enhanced e.g. by some obfuscation mechanisms [WHI05] (see 4.3).

Our goal is to work with contextual information, try to find how much private information can be derived from it and of course what are the possibilities in preventing others from deriving such information (by using PET). The second goal is to use this behaviour patterns to identify an individual among the others – this issue is closely related to so-called *linkability* (see 3.2).

Another use of these user profiles can be for intrusion detection systems. Profiles are built from normal usage data and deviations from these profiles can indicate intrusions to the system [YD02].

Discovering valuable information in huge databases involves several phases such as data preprocessing and cleaning (noise is removed); data transformation into some common structure; data mining techniques to identify interesting patterns or correlations among parts of data; interpretation of the results such as visualization or some kind of user friendly format.

3.1.1 Categorization of context information

There is a need for some context categorization, in [RDN05] two basic approaches are mentioned:

- *Conceptual viewpoint* – the type of context information we are working with (types of services, time, types of messages, ...). Conceptual cat-

egorization is a descriptive viewpoint of the contextual space (actions and relations between them).

- *Measurement viewpoint* – the actual values for a particular user or an individual (e.g. how often a service is being used, how many messages were sent, in what time a user is likely to send a message, ...).

Another approach is to consider static and dynamic context as something which is invariant (e.g. userID) and something with frequent changes (e.g. location, types of services).

3.1.2 Quality of context information (QoCI)

Quality of context information is also of a great importance because the quality of a given contextual information will significantly impact the decisions made by the autonomous system. [RDN05] define *QoCI* in terms of information quality parameters and information quality indicators:

An information quality parameter is a qualitative or subjective dimension by which a user evaluates context information quality.

An information quality indicator is a context information dimension that provides objective information about the context.

An information quality attribute is a collective term including both quality parameters and quality indicators.

An information quality indicator value is a measured characteristic of the gathered and stored data. The information quality indicator source may have an indicator value like from a sensor or user.

An information quality parameter value is the value determined for a quality parameter (directly or indirectly) based on underlying quality indicator values. Application-defined functions may be used to map quality indicator values to quality parameters values.

Information quality requirements specify the indicators required to be tagged, or otherwise documented for the information related to an application or group of applications. If a context model includes this then it is possible to make the context aware system more efficient and effective.

3.2 (Un)linkability

Unlinkability is closely related to user profiling and context modelling. The definition from Common Criteria[1] [Boa05] (see 2.4.2) is that unlinkability ensures that a user may make multiple uses of resources or services without others being able to link these uses together. PATS model (4.1.1.1) aims to evaluate the attacker's ability to link several entities together and helps to learn (or infer) some "private" information about them. It is obvious that if a system ensures unlinkability of its entities then some level of privacy is ensured. Unlinkability is one of four properties (as defined in Common Criteria) that must be satisfied in order to ensure privacy protection (these other properties are anonymity, unobservability and pseudonymity).

This section is dedicated to current context information models, behavioral models and other approaches for context information or behavioral data processing.

3.3 Common Criteria revisited

Common Criteria privacy families are defined in an existential manner and any formal definition of them has to tackle a number of ambiguities. It is unrealistic to assume perfect/absolute privacy as demonstrated by several anonymity metrics, based on anonymity sets (number of users able to use a given resource/service in a given context) [PH01] or entropy assigned to a projection between service and user/subject identities (uncertainty about using a service) [SD02b].

Can we introduce more formal definition of privacy notions and use them to define mutual relations? It is not easy, but the prospects of getting a clearer picture of mutual relations between different privacy aspects/qualities are encouraging.

Our proposal for the CC model privacy formalisation is based on the following graphical representation (fig. 3.1). 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

[1] http://www.commoncriteriaportal.org/

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 [Bel97]. Let $\mathcal{A}$ be any attacker with unbounded computing power.

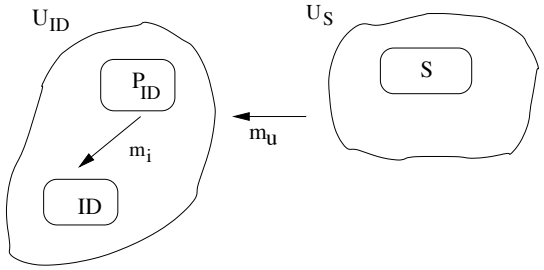


Figure 3.1: Schematics for the CC view of privacy.

A formal transcription of existential definitions of CC privacy families may be as follows.

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 1/2 whether a particular $s \in S$ or $s \in (U_S - S)$.

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. 3.1).
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 $u_{ID} \in ID$.

These existential expressions can then be easily turned into probabilistic ones that allow for expressing different qualitative levels of all these privacy concepts/families. This can be done simply by changing the "not significantly greater than" expression to "not greater than Δ", where Δ is the given probability threshold.

3.3.1 Unlinkability – The Unlinkables

Unlinkability cannot be satisfied without other privacy families. It is now understood [RI00a, RI00b] that the Common Criteria definition of unlinkability is not supporting some aspects of unlinkability in real systems, and a Common Criteria modification proposal in this manner is currently submitted. We point the reader to the fact that when pseudonymity is flawed, an attacker may obtain the ID of an actual user. The same holds when anonymity is breached.

Moreover, we are convinced that unlinkability may be a property of other privacy families. This comes straight from the formal unlinkability definition as stated above, where mapping m is the link binding the families together. Unlinkability should ensure that the particular family (or rather its implementation) does not contain side-channels (context information) that could be exploited by an attacker. We have found, in this context, two other meanings for unlinkability during our analysis. The first meaning is expressed in the following definition of unlinkable pseudonymity. It says that when a user employs two different pseudonyms, any $\mathcal{A}$ is not able to connect these two pseudonyms together.

Unlinkable pseudonymity – As for the definition of pseudonymity above

in part 3.3, and also for any $s_1, s_2 \in S$, where $s_1 \neq s_2, m_u(s_1) = u_1, m_u(s_2) = u_2$ (where $u_1, u_2 \in P_{ID}$)

1. if $\mathcal{A}$ knows ID – she cannot find (with a probability significantly greater than $1/|ID|$), whether $m_i(u_1) = m_i(u_2)$, or
2. $\mathcal{A}$ does not know ID – she cannot guess with a probability significantly greater than $1/4$ whether $m_i(u_1) \times m_i(u_2)$ belong to $ID \times ID$, $ID \times \overline{ID}$, $\overline{ID} \times ID$, $\overline{ID} \times \overline{ID}$, respectively. ($\overline{ID} = U_{ID} - ID$)

The second semantics is built on the assumption that knowledge of several pieces of mutually related information is much more powerful than knowledge of just one piece of such information. When compared with the previous definition of unlinkable pseudonymity, the definition is now concerned with a property ensuring that there is no increase in the probability of correct identification of a given user when more information is available. The same reasoning lies behind the following definition of unlinkable anonymity.

Unlinkable anonymity – As for the definition of anonymity above in part 3.3, and

1. If $\mathcal{A}$ knows ID – she cannot find with a probability significantly greater than $1/|ID|$ such $s_1, s_2 \in S$, where $s_1 \neq s_2, m_u(s_1) = m_u(s_2)$.
2. $\mathcal{A}$ does not know ID – with a probability not significantly greater than $1/4$ whether $m_u(s_1) \times m_u(s_2)$ belong to $ID \times ID$, $ID \times \overline{ID}$, $\overline{ID} \times ID$, $\overline{ID} \times \overline{ID}$, respectively.

We can apply profiling when unlinkability is breached. Basically, unlinkability should ensure that the particular family (or its implementation) does not contain side-channels that could be used when several service invocations appear.

The example: The figure 3.2 depicts how CC models our example from part 2.4.1. It is obvious that there is no information about the context information for the basket (chat) contents. This implies that an attacker will not find any link between payment cards (pseudonyms) using this model, even though the link/connection exists. This shows that CC do not address contextual information.

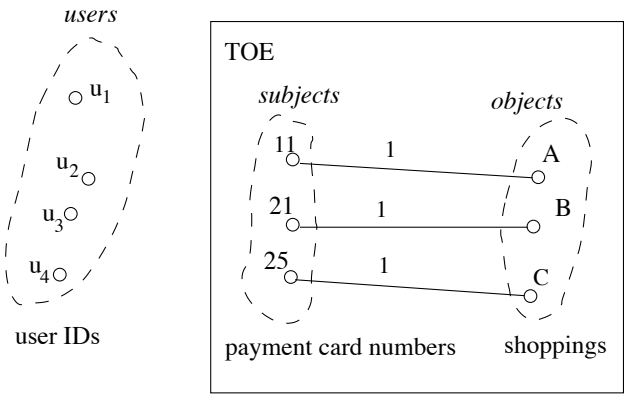


Figure 3.2: The example in the CC model.

Chapter 4

Context information, user behaviour and privacy models

In this chapter we provide an overview of some existing context modelling approaches. Good context models are essential for building context-aware applications that intelligently adapt to different environments and user tasks.

4.1 Context information models

4.1.1 Freiburg Privacy Diamond

FPD is a semiformal anonymity (and partly also unlinkability) model by A. Zugenmaier et al. [Zug03, ZKM03a]. The model originated from their research in the area of security in mobile environments. The model is graphically represented as a diamond with vertices User, Action, Device (alternatives for CC's user, service, and subject), and Location (fig. 4.1). The main reason for introducing *location* as a category here is probably due to the overall focus of this model on mobile computing.

The anonymity of a user u performing an action a is breached when there exists a connection between a and u. This may be achieved through any

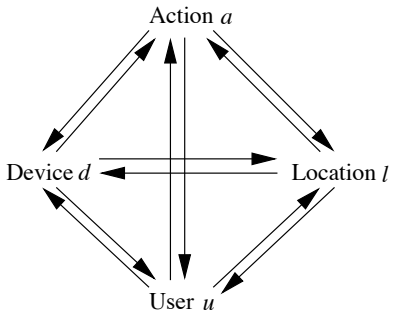


Figure 4.1: Freiburg Privacy Diamond.

path in the diamond model. Let us recap basic definitions of the FPD model:

1. Any element x has got a type $type(x) \in \{User, Action, Device, Location\}$. Any two elements, such as $x, y \in \{e | type(e) = User \vee Action \vee Device \vee Location\}, type(x) \neq type(y)$ are in a relation R if the attacker has evidence connecting x and y.
2. An action is anonymous if $U_R = \{u \mid type(u) = User \wedge (u, a) \in R\}$ is either empty or $|U_R| > t > 1$, where t is an anonymity threshold defining minimum acceptable size of anonymity set.
3. There is the transitivity rule saying that if $(x, y) \in R$ and $(y, z) \in R$, and $type(x) \neq type(z)$, then $x, z \in R$.
4. The union of all initial relations known to an attacker $\mathcal{A}$ defines his initial view $View_{\mathcal{A}}$.
5. The transitive closure $\overline{View_{\mathcal{A}}}$ of $View_{\mathcal{A}}$ defines all the information an attacker $\mathcal{A}$ may infer from her initial view.

The book [Zug03] also introduces three types of attacks with context information.

- Recognition attack – $\mathcal{A}$ realises that several users (x_i, $type(x_i) = User$) are in fact a single user.
- Linking attack – $(x, y) \in R$ and $(z, y) \in R$ are in the $\overline{View_{\mathcal{A}}}$. When $\mathcal{A}$ is able to find just one pair $(y, x_i) \in R$ then she will know that $x_i = x$ and $(z, x) \in R$.

- Intersection attack – $\mathcal{A}$ knows anonymity sets for several actions. When she knows that a certain user is in all anonymity sets, she can apply intersections to reduce size of anonymity set and eventually identify the user.

Finally, the model assigns probabilities to edges in order to express attacker's certainty about existence of particular relations with some simple rules how to derive certainty for transitive relations.

The example: When attempting to model the example scenario (see part 2.4.1) in the FPD model, the attacker ends up with three diamonds for each service use (see fig 4.2). Here *user* and *location* represent domains with no particular values as there is no such information available. The attacker cannot find any intersection of the three diamonds – i.e., there is no attack as defined by the FPD model theory. This is obvious since the FPD model does not cover any other contextual information, only location and device.

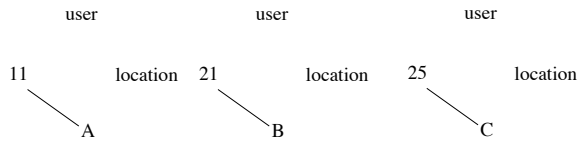


Figure 4.2: The example in the FPD model.

The FPD model only briefly mentions context information but does not introduce any definition of it. The attacks based on context information do not say how to perform them but only defines changes in $\overline{View_{\mathcal{A}}}$ when an attack is completed.

Since the FPD model newly addressed the mobile computing environment, as opposed to the old-fashioned "static" environment, location had a very prominent role, as did the device to some extent. We have decided to treat these as "ordinary" context information, i.e. as any other additional information about the system that can link a user and an action (or more precisely, their identifiers).

4.1.1.1 Context revisited – basics of the PATS (Privacy Across-The-Street) model

We propose the following approach, inspired by the way location and device (descriptors) are represented in FPD.

We suggest that all context information available to an attacker is represented as vertices in a graph, where edges are weighed with the probability of the two incident vertices (contextual information, user and service IDs) to be related/connected. Those connections may be between any two vertices, and a path connecting a user ID and a service ID with a certain probability value of the path suggests a link between the service use and the user ID exists.

The graph reflects all knowledge of an attacker at a given time. Attackers with different knowledge will build different graphs for a system as will likely do the same attacker over some time.

What is not clear to us at the moment is the question whether pseudonyms should be treated differently from other contexts or not. Clearly they are more important in the model since their connection to users and actions defines level of pseudonymity achieved in the system. Yet at the moment we suggest all vertices to be treated equally, although we suspect that some of them might be more equal than others.

4.1.1.2 Outline of the graph model

We denote the set of all vertices by V, the set of all identifiers of service instances by S, and the set of all user IDs by ID. There are no edges between any pair of elements of ID, only indirect paths through a linking context, and the same applies to elements of S. There is also a function W_{max} calculating overall probability weight for a path in the graph, and therefore also a way to determine the highest value $W_{max}(v_a, v_b)$ for a path between v_a and v_b. The value of any path is calculated as a multiplication of the weights (w) of all its individual edges, e.g. for the path $P = v_1, v_2, \ldots, v_i$ of i vertices of the graph, the value of the path P is $W(v_1, v_i) = w(v_1, v_2) \times w(v_2, v_3) \times \ldots w(v_{i-1}, v_i)$.

The following definitions also use ε to express negligible numbers (see [Bel97] for details). Let $\mathcal{A}$ be any attacker with unbounded computing power.

Unobservability (of service s_i) – a graph that $\mathcal{A}$ can build after observing a system at a given time does not include s_i at all.

Unlinkability (between two nodes v_1, v_2, at the level Δ) – a graph that $\mathcal{A}$ can build when observing the system at a given time has no path connecting v_1 with v_2 with the overall probability greater than Δ, i.e. provides $W(v_1, v_2) \leq 1/|V| + \Delta$, where $v_1, v_2 \in V$.

$W(v_1, v_2) \leq \frac{1}{|V_1.V_2|} + \Delta$, where $v_1 \in V_1$ and $v_2 \in V_2$.

Anonymity (of a user $u_{ID} \in ID$, at the level Δ) – then $\forall\, v \in V$, when $\mathcal{A}$

1. knows the set ID, she can only find a path from v to u_{ID} with the weight not greater than $1/|ID| + \Delta$, such that $W_{max}(v, u_{ID}) \leq 1/|ID| + \Delta$;
2. does not know anything about ID (particular elements or size), she can only find a path from v to u_{ID} with the weight not greater than Δ, i.e. $W_{max}(v, u_{ID}) \leq \Delta$.

Pseudonymity (of a subject/pseudonym $u \in P_{ID}$, at the level Δ) – there exists a path known to $\mathcal{A}$ from any $s \in S$ to u with a satisfactory value of $W_{max}(s, u)$, but for $\mathcal{A}$ there is no knowledge of an edge from u to any $u_{ID} \in ID$ such that when $\mathcal{A}$

1. knows ID, the path from u to any u_{ID} has weight not greater than $1/|ID| + \Delta$, i.e. $W_{max}(u, u_{ID}) \leq 1/|ID| + \Delta$;
2. does not know anything about ID (particular elements or size), the path from u to u_{ID} has weight not greater than Δ, i.e. $W_{max}(u, u_{ID}) \leq \Delta$.

There are several proposals for formal frameworks for anonymity [HO05, HS04a] and unlinkability [SK03a]. Frameworks introduced in these papers define typed systems with several defined categories like agents, type of agents, messages [HS04a] or an inductive system based on modal logic of knowledge [HO05]. We believe that our proposal would be more flexible and would cover context information as an inherent part of the model thus opening interesting questions.

The example: Let us express our example from part 2.4.1 in the PATS model. Figure 4.3 shows how the context information about typical basket

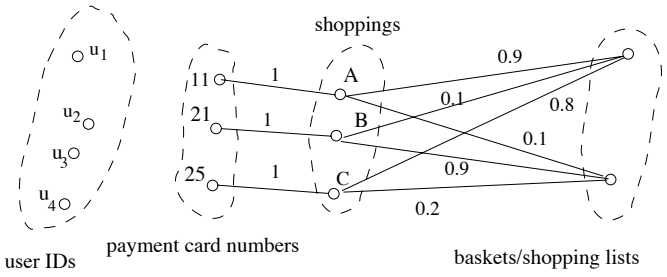


Figure 4.3: An example with a PATS model graph, where the first (upper) basket/shopping list is denoted as l and the second one as m in the table below.

contents is connected to actual instances of shoppings. As we are interested in connections between payment cards (pseudonyms), we are looking for paths (and their aggregate values) containing pairs of particular payment cards. Let us try to find paths between card 11 and the other two cards.

Path	Probabilities	Aggregate
$11 - A - l - B - 21$	$1 * 0.9 * 0.1 * 1$	0.09
$11 - A - l - C - 25$	$1 * 0.9 * 0.8 * 1$	0.72
$11 - A - m - B - 21$	$1 * 0.1 * 0.9 * 1$	0.09
$11 - A - m - C - 25$	$1 * 0.1 * 0.2 * 1$	0.02
...	...	...

Table 4.1: Paths connecting payment card 11 with the other two cards

These are the shortest (and highest value) paths only. The attacker may deduce (with a high probability) that payment cards 11 and 25 belong to the same person, though she does not know who this person is. According to our definitions, unlinkable pseudonymity is breached.

4.1.2 Categorization and modelling of quality in context information

[RDN05] provides a detailed context information categorization and discusses existing context models which will be briefly presented below. These context models are based on the following methods: set theory, directed graph, first-order logic. [**?**] also provides a good survey of context modelling approaches.

4.1.3 Set theory

[SAT+99] used set theory approach to describe context. Each context T is described by a set of two-dimensional vectors where each vector h consist of value v which describes the situations and p which indicates the certainty that the user is currently in this situation.

Set theory describes context information in a schematic way but without any indication of dependency relations.

4.1.4 Directed graph

[**?**] proposed an object-based context modelling (something like UML) in which context information is structured around a set of entities, each describing a physical or conceptual object such as person or communication channel. They use the form of a directed graph for representing context in a way where entities and attribute types are nodes and associations between them are described as edges. This model is quite comprehensive and also includes QoCI – Quality of Context Information 3.1.2 and dependency relations but not with a high accuracy. Example of this approach is on figure 4.4. We think that this approach is very descriptive, but seems to be impractical for finding behavioral patterns in very large datasets.

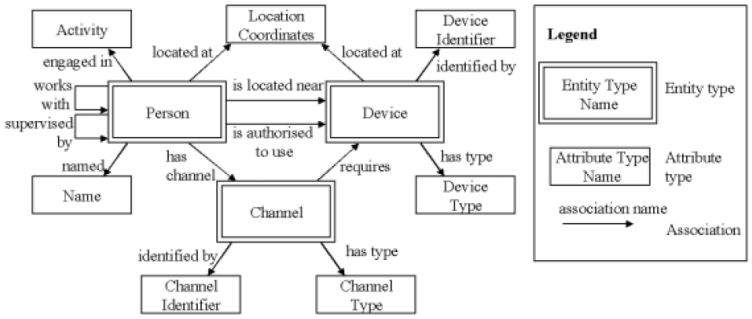


Figure 4.4: Example of the directed graph model [**?**]

4.1.5 First-order logic

[RCRM02] proposed a context model (ConChat) based on first-order predicate calculus and Boolean algebra. It covers wide variety of context information and supports couple of operations such as conjunction and disjunction of contexts as well as some quantifiers ($\exists$ and $\forall$). It provides complex first-order expressions and creation of various rules, theorems proving and queries evaluation. Context is represented with four arguments: `Context(<ContextType>, <Subject>, <Relater>, <Object>)` where *ContextType* is the type of context; *Subject* is person, place or thing, with which the context is concerned; *Object* is a value associated with the subject and *Relater* is a comparison operator (such as $=$, $<$ or $>$), verb, or preposition. Example might be something like `Context(Location, Chris, Entering, Room 3231)` or `Context(People, Room 22, >=, 3)`.

This well defined modelling technique can be used in a specific field like electronic chat (which is presented in the paper) but there is no possibility to express relations between data and the model does not deal with Quality of Context 3.1.2.

4.1.6 modelling context information with ORM

[HIM05] use a variant of Object-Role modelling (ORM), they call it Context modelling Language (CML) to model context information for supporting the development of context-aware applications. ORM approach does not address some context modelling specific problems such as *distinguishing information from different sources*, *modelling temporal data and constraints*, *modelling information quality*, *modelling information ownership*. The proposed CML addresses most of these problems. Authors present following extensions:

- source annotations, alternative fact types,
- temporal fact types,
- quality annotations,
- ownership statements,
- relational mapping

- querying and interpretation.

They used CML to produce context models for several context-aware communication applications. [HWMI05] present extensions to the model discussed above that address the challenges of assigning ownership to context information and enable users to express privacy preferences for their own information and the conditions under which this information can be disclosed to applications that request it.

4.2 User behaviour models

modelling user behaviour is an important issue for services (mainly web) that want to offer some level of customization. Users of such a services build their behavioral schemes that are used for personalization i.e. users will get different search results based on their previous activity and behaviour. Behaviour model is a probabilistic model describing which action the user will probably perform in the future. We provide an overview on existing modelling techniques with some basic description below.

4.2.1 Dynamic vs. static behavioral models

Dynamic models model some temporal variations in behaviour, which are essential for differentiation between abnormal and normal behaviour. Static models, on the other hand, do not explicitly model these temporal variations in behaviour. These models could be used for anomaly detection [YD02]. Hidden Markov Models (HMM) are used for dynamic behavioral models [HDSB03] and occurrence frequency distributions are used for static behavioral models [YD02].

4.2.2 Global mixture model

The motivation for this model is that most of personalization algorithms suffer from insufficient input data. It is impossible to learn reliable predictive profiles if a user is seen only in one or two sessions. The approach proposed by [MPG03] is to use a global mixture model to capture specific patterns

of general behaviour of the users, and once the global model is learned, the weight of each component is optimized for each known user individually. In other words the global model is personalized with individual irregularities of different users.

4.2.3 Maximum entropy model

This model and the one below are very often used for behaviour modelling and for predicting user actions (mainly on web sites) in the future [HDSB03]. Maximum entropy model provides a framework to combine information from different knowledge sources. Each knowledge source inserts a set of constraints on the combined model. The intersection of all the constraints contains a set of probability functions satisfying all the conditions. Maximum entropy principle chooses among these functions the one with the highest information entropy. [MPG03] show that the maximum entropy model outperforms Markov mixture models in recognizing complex user behaviour patterns. The latter approach uses a global model which is then optimized individually for each known user (see 4.2.2 and 4.2.4).

4.2.4 Hidden Markov Model

Hidden markov models (HMM) are stochastic models of sequential data. Each HMM contain a finite number of unobservable states. State transitions are governed by a stochastic process to form a Markov chain. At each state, some state-dependent events can be observed. The emission probabilities of these observable events are determined by a probability distribution, one for each state. Fully-connected HMMs allow state transitions between all state pairs [YD02]. To estimate the parameters of an HMM for modelling normal behaviour, sequences of normal events ([YD02] works with shell commands) collected from normal system usage are used as training examples and using an *expectation-maximization* algorithm [DLR77] the maximum-likelihood parameters are estimated.

4.3 Privacy models

The need for privacy protection is obvious. It is possible for an eavesdropper to infer private information about the user e.g. from his requests on the Internet, web sites he accessed and linkage between these sites. There are some privacy model proposals to deal with privacy issues mainly of web sites users. [ESM02b] propose an approach to confuse the eavesdropper's automated programs with wrong data. Their model is based on the generation of faked transactions in various fields of interest in order to prevent the eavesdropper from accurate derivation of the user profile. They also propose a privacy measure that reflects the degree of confusion a system can cause to the eavesdropper. This approach can also be used for hiding the information interests of a homogenous group of users who share a local area network and an access point to the Web [ESM02a].

Freiburg privacy diamond [ZKM03b] is a model proposed for security in mobile environments. The model consists of four vertices User, Action, Device, Location and edges between them. The goal of the attacker then is to connect a User with an Action. For expressing attacker's certainty about relations in the model, edges can be assigned with some probabilities.

The above model is in [MC04] compared with a new proposed graph model (PATS – Privacy Across The Street) which aim is to include contextual information for more precise description of entities. Vertices in the graph represent all available contextual information and edges (that are also weighted with probabilities [CKM06, CKM05]) represent relation between vertices.

4.4 Data mining

When we discuss privacy issues, one should also mention data mining. Data mining techniques are used for searching large volumes of data looking for patterns and various data relationships. It encompasses various techniques like association rules, cluster analysis (see 4.5), decision trees, neural networks, genetic algorithms and exploratory data analysis.

It is important to discuss how data mining can violate personal privacy. "Proper" use of data mining techniques can lead to some private data inference. [Tav99] provides a comparison between "traditional" retrieval of

personal information and data mining approaches and [CM96] discuss security and privacy implications of data mining. [Bro00] discuss two views on data mining – the desire for privacy by web users and the need for web content providers to collect and utilize data about users – users may be unaware how much identifying information can be disclosed; and (from the point of web content providers) how privacy enhancing technologies (PET) can substantially invalidate data mining results.

Data mining is widely used for discovering web users' navigational characteristics and patterns for better understanding of their needs and for providing some levels of customization [BBA+00, BL00, DIC00].

4.5 Cluster analysis

Cluster analysis (as one of the data mining techniques) is a statistical tool for grouping object into distinct sets. For our purposes, clustering can be successfully used for pattern finding – mainly web users behavioral patterns [CHM+00, Lek00]. There are three main properties characterizing this approach:

- Cluster analysis encompasses a number of different algorithms and methods for grouping objects of similar kind into respective categories.
- Cluster analysis aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise.
- Cluster analysis can be used to discover structures in data without providing an explanation or interpretation.

A good overview on various clustering data mining techniques can be found in [Ber02].

4.5.1 Measuring distance between objects

There are several methods for measuring distance between objects. We mention only the three most important ones [JW02].

- **Joining (Tree Clustering)** is a method that joins objects together into successively larger clusters, using some measure of similarity or distance. Typical result of this type of clustering is the hierarchical tree. This approach is efficiently used by [FSS00] for clustering web usage sessions.
- **Two–way Joining (Block Clustering)** groups objects into pre-clusters which are then treated as single cases. Standard hierarchical clustering is then applied to the pre-clusters in the second step.
- k–**Means Clustering** forms exactly k clusters that are as distinct as possible.

We need some criteria for differentiating objects and forming clusters. There are metrics for both single and multi dimensional objects. The most common way for multidimensional objects is *Euclidean distance* which is geometrical distance between two points in multidimensional space:

$$distance(x, y) = \sqrt{\sum (x_i - y_i)^2} \tag{4.1}$$

Some other commonly used metrics are:

- *Squared Euclidean* in comparison to euclidean distance puts more weight to isolated objects.

$$distance(x, y) = \sum (x_i - y_i)^2 \tag{4.2}$$

- *City-block (Manhattan)* often provides similar results as euclidean distance but restricts the impact of outlying objects (outliers).

$$distance(x, y) = \sum |x_i - y_i| \tag{4.3}$$

- *Chebychev* is suitable for situations where we want to differentiate objects that are different e.g. in only one dimension.

$$distance(x, y) = \max |x_i - y_i| \tag{4.4}$$

4.5.2 Hierarchical clustering – measuring distance between clusters

This method takes the matrix of distances and creates hierarchical sequence of decompositions in a way that we start with n clusters where each of them consist of only one object. Then we try to find two clusters whose distance is minimal and put them together. We will end up with one large cluster. To make this algorithm working we need to define how the distance between two clusters is measured [JW02].

- *Nearest neighbour* – Distance between two clusters is the minimal distance between two objects where each of them is from different cluster. Disadvantage of this approach is that far objects are often put to the same cluster.
- *Furthest neighbour* – The dissimilarity between 2 groups is equal to the greatest dissimilarity between a member of cluster i and a member of cluster j. This method tends to produce very tight clusters of similar cases.
- *UPGMA (unweighted pair group method using averages).* The distance between two clusters is the mean of distances between all possible inter-cluster pairs. UPGMA is generally preferred over nearest or furthest neighbor methods since it is based on greater information.

4.6 Different types of similarity measures

It is appropriate to encompass more similarity measures to capture the users' interests. We can get following information from a web log: the frequency of a hyper-page usage, this lists of links an user selected, the elapsed time between two links, and the order of pages accessed by individual web users. Based on this information we can respective similarity measures. These are: usage based, frequency based, viewing-time based and visiting-order based [XZJL01, XZ01]. Each of them is slightly discussed in next four subsections. For the next description we suppose that there is a given web site S, m users $U = u_1, u_2, \ldots, u_m$ who accessed n different web pages $P = p_1, p_2, \ldots, p_n$ in some time interval. *Usage value* is associated to each page p_i and user u_j.

This value is denoted as $use(p_i, u_j)$, and defined as

$$use(p_i, u_j) = \begin{cases} 1 & \text{if } p_i \text{ is accessed by } u_j, \\ 0 & \text{otherwise.} \end{cases}$$

This value can be computed from the access logs of the site.

4.6.1 Usage based measure

This similarity is measured by the number of common pages that the users accessed within a specific period of time (we are interested in what pages did the user accessed but not how frequent). The measure is defined by

$$Sim1(u_i, u_j) = \frac{\sum_k (use(p_k, u_i) \cdot use(p_k, u_j))}{\sqrt{\sum_k (use(p_k, u_i)) \cdot \sum_k (use(p_k, u_j))}} \tag{4.5}$$

where $\sum_k use(p_k, u_i)$ is the total number of pages that were accessed by user u_i and $\sum_k use(p_k, u_i) \cdot \sum_k use(p_k, u_j)$ is the number of common pages accessed by both u_i and u_j. If two users accessed same pages (exactly), their similarity will be 1 [XZJL01, XZ01].

4.6.2 Frequency based measure

We can express the similarity by counting the number of times each user accessed separate pages at all sites. The measure is defined by

$$Sim2(u_i, u_j) = \frac{\sum_k \sum_s (acc_s(p_k, u_i) \cdot acc_s(p_k, u_j))}{\sqrt{\sum_k \sum_s (acc_s(p_k, u_i))^2 \cdot \sum_k \sum_s (acc_s(p_k, u_j))^2}} \tag{4.6}$$

where $acc_s(p_k, u_i)$ is the total number of times that a user u_i accessed the page p_k at site s [XZJL01, XZ01].

4.6.3 Viewing-time based measure

Another way of expressing the similarity is by taking into account the actual time the users spent on viewing each web page. $t(p_k, u_j)$ denotes the time

the user u_j spent on viewing page p_k ($t(p_k, u_j) = 0$ if u_j did not access page p_k). The similarity between users is then expressed by

$$Sim3(u_i, u_j) = \frac{\sum_k (t(p_k, u_i) \cdot t(p_k, u_j))}{\sqrt{\sum_k (t(p_k, u_i))^2 \cdot \sum_k (t(p_k, u_j))^2}} \tag{4.7}$$

where $\sum_k (t(p_k, u_i))^2$ is the square sum of the time the user u_i spent on viewing pages at the site, and $\sum_k (t(p_k, u_i) \cdot t(p_k, u_j))$ is the inner-product over time spent on viewing the common pages by users u_i and u_j. Even if the accessed pages are the same for two users, their similarity might be less than 1 due to different amount of time they spent on the pages [XZJL01, XZ01].

4.6.4 Visiting-order based measure

Last measure is based on visiting order and r-hop navigation paths $Q = q_1, q_2, \ldots, q_r$. The similarity between path Q^i and Q^j for user i and j, respectively is then computed using the natural angle (i.e. *cos*) [XZJL01, XZ01].

Authors then use a clustering algorithm to differentiate users into distinct groups. This is based on one of the similarity measure described above. The clustering techniques and algorithms are described in section 4.5.

Chapter 5

Metrics

In general, designing reasonable metrics for privacy quantification is a multidisciplinary approach, as pointed out in previous chapters. This chapter focuses on technical and formal approaches. These approaches can be distinguished depending on purposes or use-cases, available data, and the way results can be interpreted.

The purpose of a privacy metric can be to measure the degree of privacy which a protocol or a communication system can provide to its users. Another purpose can be to provide a metric for the privacy which a user actually may expect with respect to her situation, that is, for instance, her previous actions.

Data can be available as persistent data, maybe organized in a database. There are lots of cases in which organizations carry out so-called anonymous surveys. If anonymity is understood as cutting off the name and address only, it is hard to estimate whether the remaining attributes are not sufficient to re-identify individuals. Privacy metrics help to assess the significance of single attributes with respect to re-identification.

Another possible data source can be a set of observations. In contrast to a database, such a set of observations covers actions or events which occurred in a network over time. Observations do not necessarily need to be complete with respect to these actions or events. However, the more complete the observations are the better can privacy be assessed.

The results of the metrics can be distinguished into possibilistic and probabilistic measures, and additionally in worst-case and average-case approaches. Worst-case approaches may be too strict in some cases. In fact, a system which provides no anonymity in the worst case may work well in the majority of other cases. However, the opposite is also a flaw of average-case approaches, since a system may provide appropriate anonymity in average, while still failing in important situations.

Possibilistic measures deal with anonymity sets directly. If subjects belong to the set, they are considered to be anonymous. The greater the set appears for an adversary the greater is the anonymity of subjects within the set. This leads to discrete results, however, possibilistic measures restrict the model to exactly one view of the world. Probabilistic measures, in contrast, deal with entropies, which are borrowed from information theory. The entropy of an observed attribute value, conditional to the adversary's prior knowledge, yields the degree of information which an adversary is able to gain from her observation. Thus, the entropy can also be used to estimate the size of the anonymity set which remains after the adversary's observation.

In the course of different traditions, several approaches have been developed for privacy metrics. In this chapter, we describe approaches in the field of formal methods (Section 5.1), surveys and statistical databases (Section 5.2), and data-flow analysis in networks (Section 5.3). In Section 5.4, we outline generalizations of the previous approaches and refer to work which is currently state of the art.

5.1 Formal Methods

There have been several proposals for analyzing anonymity properties in a formal manner. First of all, these approaches measure the degree of anonymity as in discrete values, that is either (a flavor of) anonymity is preserved or not. For most approaches, the authors state that probabilistic enhancements would be possible. In this section, however, we focus on the possibilistic foundations without any enhancements. In these approaches, the determination of the degree of anonymity is done by so-called function views which will be described first or by using a semantic characterization of anonymity in formal languages. The latter will only be outlined in this section.

5.1.1 Function Views

For specification of information-hiding properties, Hughes and Shmatikov utilize the concept of function views [HS04b]. Supposed, the capability of an adversary to obtain data is modeled by functions. For instance, a function $s : \mathcal{M} \to \mathcal{A}$ assigns a sender (from a set of subjects $\mathcal{A}$) to a conversation (from a set of conversations $\mathcal{M}$). Then, information-hiding properties can be expressed in a straight-forward manner by restricting the adversary's knowledge about function s. Hughes and Shmatikov point out that it is sufficient for information-hiding to restrict three properties of functions, that is the graph, the image, and the kernel. These restrictions determine the view which an adversary has on the function.

The graph of a function $f : \mathcal{A} \to \mathcal{B}$ is the corresponding relation graph $f \subseteq \mathcal{A} \times \mathcal{B}$ which consists right of these tuples (a, b) for which $f(a) = b$ holds with $a \in \mathcal{A}$ and $b \in \mathcal{B}$. That is

$$\operatorname{graph} f = \big\{(a, b) \;\big|\; f(a) = b\big\} \tag{5.1}$$

For instance, for

$$f(x) = \begin{cases} 1 & \text{for } x = 1 \\ 3 & \text{for } x = 2 \\ 3 & \text{for } x = 3 \end{cases} \tag{5.2}$$

we achieve $\operatorname{graph} f = \{(1, 1), (2, 3), (3, 3)\}$.

The image of a function $f : \mathcal{A} \to \mathcal{B}$ consists of these elements $b \in \mathcal{B}$ for which there is an $a \in \mathcal{A}$ such that $f(a) = b$. Note that there may be elements $b' \in \mathcal{B}$ for which there is no $a \in \mathcal{A}$ that satisfies $f(a) = b$. Formally, we denote

$$\operatorname{im} f = \big\{f(a) \;\big|\; a \in \mathcal{A}\big\} \tag{5.3}$$

Continuing the example of Equation 5.2, we achieve $\operatorname{im} f = \{1, 3\}$.

The kernel of a function $f : \mathcal{A} \to \mathcal{B}$ consists of equivalence classes. These equivalence classes consists of all elements $a \in \mathcal{A}$ for which f maps to one and the same $b \in \mathcal{B}$. That is,

$$\langle a, a' \rangle \in \ker f \iff f(a) = f(a') \qquad \text{with } a, a' \in \mathcal{A} \tag{5.4}$$

Continuing the example of Equation 5.2, we achieve $\ker f = \{\{1\}, \{2, 3\}\}$.

A function view can then be denoted as a triple $\langle F, I, K \rangle$ where $F \subseteq \mathcal{A} \times \mathcal{B}$ describes the knowledge about the graph, $I \subseteq \mathcal{B}$ describes the knowledge about the image, and K is the equivalence relation on $\mathcal{A}$ which describes the knowledge about the kernel of f. In order to let the view $\langle F, I, K \rangle$ be restrictive with respect to f, the following constraints need to be satisfied:

- $F \supseteq f$, that is the graph of the view leads to the same or more uncertainty about the actual relation of inputs and outputs of f.
- $I \subseteq \operatorname{im} f$, that is the image of the view supports all or less many different outcomes of f.
- $K \subseteq \ker f$, that is the kernel of the view is still a sound part of the kernel of f, however, the outcome of f depends on the same or less many equivalence classes of input values.

Using the notion of function views, function constraints, or flavors of opaqueness [HS04b], can be systematically expressed in terms of first-order logic. However, Hughes and Shmatikov also point out that F, I, and K are not independent from each other.

Suppose, for instance, the function $s : \mathcal{M} \to \mathcal{A}$ maps conversations (elements of $\mathcal{M}$) to senders (elements of $\mathcal{A}$). Kernel opaqueness, that is two arbitrary inputs of f can necessarily be mapped to different outcomes, leads to untraceability. An adversary is then not able to relate two conversations unambiguously to each other, even if they have been sent by the same sender.

Hughes and Shmatikov furthermore show how function views can mediate between system specification which are commonly done in process algebras and (information-hiding) property specifications which are commonly done in some logic.

5.1.2 Formal Languages and Semantics

Halpern and O'Neill [HO03] address the topic of mediation and show that, though elegant and useful, function views are not necessary for mediation, since all the specification can be done by semantic characterizations.

Schneider and Sidiropoulos [SS96] use the modelling language CSP (Communicating Sequential Processes) for a process algebraic formalisation of anonymity. Syverson and Stubblebine describe anonymity properties in formal languages based on group principals [SS99]. They describe the information which is to be protected and the purpose of the protection, i.e. the degree of anonymity.

In a former approach [Shm04], Shmatikov formalizes the Crowds model [RR98] of Reiter and Rubin by means of Markov chains and describes the degrees of anonymity of [RR98] in temporal probabilistic logic formulas.

5.2 Persistent Data & Statistical Databases

The question of privacy in databases of personal data records was tackled in large-scale when it came to the census discussion in (Western) Germany during the 1980s. Fischer-Hübner [FH87, FH01] points out that such data records consist of three kinds of data, that is *identity data*, *demographic data*, and *analysis data*. With identity data, it is possible to identify distinct persons, and thus, this data is obviously privacy-relevant. This could be, for instance, name or address. However, Fischer-Hübner also shows that the common assumption at that time, that truncating identity data would lead to anonymous data records, does not hold. It is rather obvious that combinations of demographic data, such as sex, nationality, education, religion, and marital status, can be used to re-identify people. Therefore, these items are also privacy-relevant.

There have been several complementary approaches [Rub93, BKP90], one of the most famous in the privacy-community has been proposed by Sweeney. In [Swe02b], she points out that classical access control approaches fail to protect against data disclosure. This particularly is the case, if protected data is not subject of the release process, but results from the derivation of legitimately released data. Therefore, at least unambiguous relations between released data and supplementary knowledge must be avoided.

The actual threat which arises from database contents depends on the recorded attributes and the frequency distribution of their values. Fischer-Hübner proposes a probabilistic approach for assessing the *risk of re-identification*, whereas the *k-anonymity* approach proposed by Sweeney is possibilistic. Therefore, *k-anonymity* is only applicable for worst-case considerations,

whereas Fischer-Hübner's approach yields average-case results.

Both approaches require quite strong assumptions. The data-holder (or whoever wants to assess the threats of re-identification) has to determine the quasi-identifier[1] properly, in any case. The assumption is that she does so. Furthermore this person needs access to all databases which could be used for identification. In case of public databases, this requirement is satisfied. If the adversary uses supplementary data which is not publicly available, then the quasi-identifier cannot be chosen appropriately.

5.2.1 Risk of Re-identification

The approach of Fischer-Hübner [FH87, FH01] can be understood as a metric of uniqueness of attribute values (or combinations) within a database. Her approach is based on Shannon entropies [Sha48].

Suppose that there is a database table with n records. Each record consists of values for a set of discrete attributes, including $X_1, \ldots, X_m$.

Fischer-Hübner defines the risk of re-identification $r(X_1, \ldots, X_m)$ as the ratio between the *average number of value combinations* $n_{vc}(X_1, \ldots, X_m)$ that can be used for re-identification and n.

$$r\big(X_1, \ldots, X_m\big) := \min\left(1, \frac{n_{vc}(X_1, \ldots, X_m)}{n}\right) \tag{5.5}$$

We need to enforce 1 as the upper bound, since the ratio can in fact be greater than 1. This happens, if the number of such value combinations is greater than the number of records. However, a risk greater than one has no useful interpretation.

The average number of value combinations which can be used for re-identification is defined by means of the entropy $H(X_1, \ldots, X_m)$ of all involved attributes $X_1, \ldots, X_m$. That entropy yields the information which can be obtained from the corresponding values to these attributes. The entropy's dimension is Bit, thus, the average number of value combinations which contribute information is 2, that is the number of states of a single Bit, to the power of the entropy.

$$n_{vc}(X_1, \ldots, X_m) := 2^{H(X_1, \ldots, X_m)} \tag{5.6}$$

[1] The *quasi-identifier* is a set of database attributes which unambiguously identifies a subject within the database.

The greater the entropy is, indeed, the greater is the number of value combinations which can be used for re-identification. In fact, the entropy is the greater the more the frequency distribution of all involved attribute values converges to uniform distribution. Furthermore, it is the smaller the more uneven this frequency distribution is. In addition, the entropy depends also on the number of involved attributes. The greater this number is the greater is the entropy and vice versa.

Strictly speaking, the joint entropy $H(X_1, \ldots, X_m)$ is defined as a sum of conditional entropies. We do not elaborate the details in this chapter, however, more detailed background can be found in [Sha48]. By means of conditional entropies, Fisher-Hübner's approach can also take dependencies between different attributes into account.

5.2.2 k-Anonymity

Sweeney [Swe02b] proposes a possibilistic approach. It can be used to assess threats of re-identification which arise from linking attributes which are shared between different databases.

Supposed, two database tables overlap in a subset of such a quasi-identifier, then we can count the occurrences of records with the same values in this attribute subset. The actual measure which is provided by k-anonymity is k which denotes the smallest count of these occurrences.

The records from both databases cannot unambiguously be linked as long as k is greater than 1. The reverse, however, does not generally hold, since $k = 1$ only states that at least one record can be linked. In particular, it makes no difference with respect to k, if just one or many more records can be linked. The greater k is, however, the greater is the anonymity assumed to be.

5.3 Data-flow in Networks

There are several different approaches for systems and protocols which were meant to preserve the user's anonymity, for instance, the DC net, mix-based approaches, or Crowds. The efficiency of these approaches with respect to

required resources can be assessed by means of traditional analysis methods. Assessing the anonymity which they provide, however, turned out to be a more severe problem. In this section, we focus on approaches which tackle this topic.

Díaz et al. proposed a measurement [DSCP02] which assesses the sender anonymity that can be provided by a communication system. A similar approach has independently been proposed by Serjantov and Danezis [SD02a, Dan03] at around the same time. The difference between both approaches is mainly that Díaz et al. normalize the entropy, whereas Serjantov and Danezis use the entropy measure without normalization. In [Dan03], Danezis discusses the pros and cons of normalization with respect to these measurements. His conclusion is basically that, by normalization, important information about the measured anonymity gets lost, particularly the average size of the corresponding anonymity set. Díaz et al. argue, however, for a quality measurement which is independent from anonymity set size and only relies on the distribution of probabilities. That is, the probabilities of users for being the sender of a particular message.

The foundation of both approaches is Shannon's information theory [Sha48]. Assuming that adversaries are able to carry out observations and assign corresponding probabilities to possible senders of a message, there are various kinds of observations that an adversary may use, for instance results from traffic analysis, timing attacks, message length attacks, or generally information leaks of the communication system. By means of assigning probabilities, adversaries are able to distinguish possible senders of a message much better than by assigning them to anonymity sets. Entropy or the information which is contained in a given distribution of probabilities, is used to assess the information that the adversary was able to obtain. From the point of view of a user, entropy is used to assess the anonymity of the user with respect to the adversary's observation.

Díaz et al. define the information leak of the adversary's attack as the difference between maximum entropy of the system and the actual entropy of the system after the adversary's observation. Thus, by denoting the maximum possible entropy[2] as H_M and the entropy after an observation as $H(X)$, the information which the adversary has learned can be assessed by $H(X)/H_M$. Here, X is a discrete random variable with probability mass function $p_i = \Pr(X = i)$, where i is an index over all users in the system.

[2]Not to confuse with max-entropy.

The entropies $H(X)$ and H_M can be calculated as shown in Equation 5.7, the latter one by means of the number of all users in the system n.

$$H_M = \log_2(n) \qquad H(X) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{5.7}$$

The *degree of anonymity*, denoted as d, is then defined as the difference between the state of perfect anonymity and the adversary's gain of information. As mentioned, this degree is normalized with respect to H_M:

$$d = 1 - \frac{H_M - H(X)}{H_M} = \frac{H(X)}{H_M} \tag{5.8}$$

This degree is a value between 0 and 1, where 0 denotes no preserved anonymity and 1 denotes perfect anonymity with respect to the system. That is, the adversary is able to identify the user as sender of the message, in case of $d = 0$. The other extreme value $d = 1$ would mean that the adversary is just able to guess the sender, since all users appear evenly suspicious for having sent the message. This case is similar to an anonymity set which contains basically all users. And otherwise, that is d is neither 0 nor 1, it holds, the greater d is the greater is the average anonymity.

With this normalization, it is possible to assess arbitrary communication systems with respect to a given lower bound of anonymity. Díaz et al. point out, however, that such a lower bound depends very much on the system requirements and can hardly be suggested, generally.

Serjantov and Danezis [SD02a, Dan03] use $H(X)$ without any normalization to assess the anonymity. This yields the average set size of a corresponding anonymity set and, therefore, a measure about the actual effort which an adversary has to take into account for identifying a user as sender of a message. This average size k of the anonymity set can be calculated by means of the dimension of the entropy $H(X)$ which is Bit:

$$k = 2^{H(X)} \tag{5.9}$$

These approaches can easily be adapted for recipient anonymity or any other action.

5.4 Generalizations

Both measurements which have been described in the previous section are useful to quantify the effort of an adversary to compromise all messages, that is to assign the messages to users. Tóth et al. refer to this quantification as global measure [THV04a] and point out that it is of little use for users to quantify their particular anonymity. They refer to the latter quantification as local aspect of anonymity and prove that different probability distributions which provide very different local anonymity lead to the same level of anonymity with respect to global measurements. Furthermore, they show that for a given degree of anonymity there is a always a corresponding probability distribution which is not desirable for all users. Thus, Tóth et al. conclude, global measures do not provide a quantification of anonymity with respect to local aspects.

5.4.1 Local Anonymity

In order to overcome the shortcomings, they propose [THV04a] an upper bound Θ and suppose that an adversary is successful, if she can assign a message to a user with probability greater than this upper bound. Thus, a system provides sender anonymity as long as for all received messages β and all senders s holds that the probability for s being the sender of β is lower or equal Θ, formally

$$\forall\beta.\forall s.\big(P_{\beta,s} \leq \Theta\big) \tag{5.10}$$

Dually, this can be formalized for recipient anonymity.

This is a generalization of global measures, since the (global) degree of anonymity d can be assessed as well as $H(X)$ by using Θ:

$$d \geq -\log_n \Theta \qquad \text{where } n \text{ is the number of senders} \tag{5.11}$$

$$H(X) \geq -\log_2 \Theta \tag{5.12}$$

5.4.2 Towards Arbitrary Attributes

The ideas of this section mainly reflect the current work of Sebastian Clauß and will be elaborated within a greater context in his PhD thesis.

5.4.2.1 Modelling the Observer's Knowledge Base

By observing actions, an observer gets a limited insight into user's personal information (hence we address it as a set of attributes) and into relations between different attribute values. The observer can collect this information, and may conduct any desired statistical analysis on them. With a growing number of observations, the information on probability distributions of the digital identities gets more exact[3]. Clauß defines the knowledge of an attacker, which he gained by observations, in form of the *observer state*:

Definition 1 (Observer State) *The* State $Z^{\mathcal{X}}$ *of an observer* $\mathcal{X}$ *is a triple* $(\mathcal{I}, h, g)$*, where:*

- $\mathcal{I}$ *is the set of all digital identities possible.*

$$\mathcal{I} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$$

- $h : \mathcal{I} \mapsto \mathbb{R}$ *is a function, which assigns a probability to each digital identity, i.e.,* $(\forall i \in \mathcal{I}.\ 0 \leqq h(i) \leqq 1)$
- g *is the number of observations leading to this* state*.*
- *the sum of all probabilities is 1.*

$$\sum_{(\mathcal{I})} h(i) = 1$$

$h(i)$ denotes the probability that within the set $\mathcal{I}$ of all possible identities, the identity i is observed by the attacker.

When the attacker observes a user's action, the probability of the identities matching the observation (i.e., the suspects with respect to the observation) is increased, whereas the probability of all other identities is decreased. After defining observations, Clauß specifies a method for matching identities and observations.

Definition 2 (Observation) *An* observation *is a (possibly incomplete) bundle of attribute values. Such a bundle contains at most one value per*

[3] "exact" here means exact with respect to the observation. Observations may nevertheless yield incorrect information.

attribute. The set $\mathcal{B}$ of all possible observations *is the cross product of all attributes with an additional element "not observed" $\perp$.*

$$\mathcal{B} = (\mathcal{A}_1 \cup \{\perp\}) \times (\mathcal{A}_2 \cup \{\perp\}) \times \cdots \times (\mathcal{A}_n \cup \{\perp\})$$

Intuitively, this means that during actions a user discloses attribute values. The observer *observes* this values and gets a more and more refined view on the digital identities and by that on the users.

Within the set of all possible digital identities, an observer can separate suspect digital identities with respect to an observation from non-suspect digital identities. The set of *suspects* related to an observation can be defined as follows:

Definition 3 (Suspects) *The set of* suspects $\mathcal{V}_b$ *related to an observation $b = (x_1, .., x_n)$ contains all digital identities $i = (x'_1, .., x'_n)$, whose attribute values are either equal to attribute values of b or are not contained in b.* [4]

$$\mathcal{V}_b = \{i | x_k \in \{x'_k, \perp\}, k = 1, .., n\} \tag{5.13}$$

As stated above, the observer *learns* by observations. The following definition formalises this learning process:

Definition 4 (Observer State Update) *Let $b \in \mathcal{B}$ be an observation and $\mathcal{Z}$ a set of observer states. An* observer state update $\delta : \mathcal{Z} \times \mathcal{B} \rightarrow \mathcal{Z}$ *constructs a new observer state from a given state and an observation.*

These definitions are a framework for formalising concrete observations and statistical analysis based on digital identities. In order not to restrict this model to passive (observing only) attackers, it is intentionally not defined how an observation is done. So, an attacker may observe messages, but may also actively insert or fake messages in order to observe users' reactions.

Based on the above definitions, a statistical observer model is defined as follows:

[4] The matching function "equality" used here is a simple example. This makes only sense, if attribute values are discrete and not related to each other. If this is not the case, e.g., if measuring faults for originally continuous attribute values need to be taken into account, other matching functions should be used which reflect such properties of attributes.

Definition 5 (Statistical Observer Model) *A* statistical observer model *of an observer $\mathcal{X}$ comprises a set $\mathcal{I}$ of digital identities, a set of observations $\mathcal{B}$, a set $\mathcal{Z}^{\mathcal{X}}$ of observer states and a function δ, which derives new observer states from previous states and observations.*

The statistical observer model specifies the observer's knowledge in form of statistics about digital identities together with a method for aggregating newly gained knowledge. This is an abstract definition, as it leaves open how the aggregation of new observations actually influences the probabilities of digital identities.

5.4.2.2 Concrete Statistical Observer Model

Concrete Observer State Update Method In order to actually aggregate knowledge about entities within the system, we need to define a concrete observer state update method, i.e. given an observer state, how exactly the probabilities of the digital identities change upon an observation[5].

Thereby, the major goal is that, by observations, the frequency distributions of attribute values within the observer state shall converge to the actual frequency distributions within the system. Further, the model shall reflect observed relations between values of different attributes.

Given a set of digital identities $\mathcal{I}$ and the set of all observations possible $\mathcal{B}$, the concrete observer model is defined in an inductive way.

First, the initial state is defined, in which the attacker did not do any observations. For the initial state $Z_0 = (\mathcal{I}, h, g)$ it shall hold that $g = 0$ and h is uniformly distributed, i.e. $(\forall i \in \mathcal{I}.h(i) = \frac{1}{|\mathcal{I}|})$.

Now we specify how an observation actually changes the probabilities of the digital identities. A function $\delta : \mathcal{Z} \times \mathcal{B} \rightarrow \mathcal{Z}$ derives state $Z_{k+1} = (\mathcal{I}, h_{k+1}, g_{k+1})$ from the previous state $Z_k = (\mathcal{I}, h_k, g_k)$ and an observation $b \in \mathcal{B}$ as follows:

[5]The update method described here is an example, in order to show a possibility how observations can be aggregated in a meaningful way into a *statistical observer model*. There may exist other concrete models.

$$h_{k+1} \quad : \quad i \mapsto \frac{h_k(i) * g_k + x}{g_k + 1} \tag{5.14}$$

$$x \quad = \quad \begin{cases} \frac{1}{|\mathcal{V}_b|} & \text{iff } i \in \mathcal{V}_b \\ 0 & \text{otherwise} \end{cases}$$

$$g_{k+1} \quad = \quad g_k + 1 \tag{5.15}$$

This intuitively means, that first each observation is *weighted* by 1. Then, this *weight* is divided by the number of suspects of this observation. By doing that, more significant observations (i.e., observations containing values of more attributes) get a bigger influence on the probability of the suspect identities than less significant ones. Further, the *weight* of the observation is set into relation to the number of observations already aggregated, so that every observation already aggregated has the same overall influence on the probabilities.

Example: Let a model contain three attributes with two values each. Within this model, at most eight digital identities can be distinguished. Shown as a cube, each corner represents a digital identity (see Figure 5.1). The figure shows how the observer state changes by an observation $b = (0, 0, \perp)$.

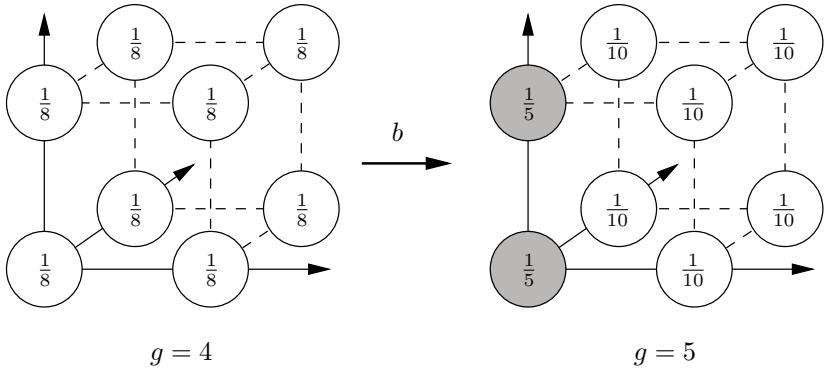


Figure 5.1: Suspects and non-suspects relative to an observation b

$\Diamond$

In fact, the observer model defined above sums up relative frequencies. With a growing number of observations, it can be assumed that the relative frequencies converge to probabilities. By induction over g, it can be shown,

that function h always has the properties of a probability distribution, i.e., $\sum_{(i \in \mathcal{I})} h(i) = 1$ and $h(i)$ is not negative.

Lemma 6 *The sum of all probabilities $h(i)$ of any single attacker state $Z = (\mathcal{I}, h, g)$ is 1.*

$$\sum_{(i \in \mathcal{I})} h(i) = 1$$

Proof: by induction over g.

□

Lemma 7 *Function $h(i)$ is not negative.*

Proof: by induction over g. □

Corollary 8 *The model of the observer knowledge base is coherent.*

Proof: We need to show, that $\forall i \in \mathcal{I}.(0 \leq h(i) \leq 1)$ holds for all possible states. This follows directly from Lemmas 6 and 7. □

Merging Multiple Observer States In the general case, it is always possible to collect observations from different sources of information, and aggregate them within a single observer state. So, partial information of different sources can be merged.

A special case for merging is the situation that two observers use the same set of attributes and values, i.e. their model bases on the same identities. In this case two observer states can be aggregated without the need to add every single observation of one state to the other. So, observer states of different sources of information can easily be aggregated into a general state.

Definition 9 (State Aggregation) *Two states $Z^A = (\mathcal{I}, h^A, g^A)$ and $Z^B = (\mathcal{I}, h^B, g^B)$ based on the same set of digital identities are* aggregated *to a new state $Z^A \cup Z^B = (\mathcal{I}, h^C, g^C)$ as follows:*

$$g^C = g^A + g^B \tag{5.16}$$

$$h^C : i \mapsto \frac{g^A h^A(i) + g^B h^B(i)}{g^C} \tag{5.17}$$

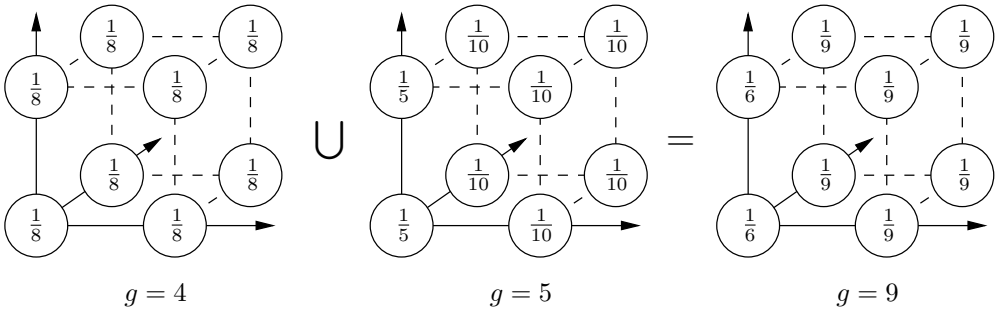


Figure 5.2: Aggregation of two observer states

Example: In Figure 5.2 aggregation of observer states is shown.

$\Diamond$

In the following, we prove correctness of this aggregation. For doing this, we first need a definition of *equal observer states*.

Definition 10 (Equal observer states) *Two observer states* $Z^A = (\mathcal{I}^A, h^A, g^A)$ *and* $Z^B = (\mathcal{I}^B, h^B, g^B)$ *are* equal *in case* $\mathcal{I}^A = \mathcal{I}^B$, $h^A = h^B$ *and* $g^A = g^B$.

Further, we need the following lemma:

Lemma 11 (Order of observations) *Let* Z *be an arbitrary observer state and* b_1, b_2 *observations. The* order of the aggregation *of these observations into* Z *does not influence the resulting state.*

Proof: Aggregation of additional observations does not change the set of identities within the observer state. As the number of observations added is the same for all orders, g is equal in all possibly resulting states. Finally, we need to show that the result of function h does not change for all possible orders of aggregation. Based on the concrete observer state update method described in 5.4.2.2 the following holds:

$$\frac{\frac{h(i)g+x_1}{g+1}(g+1)+x_2}{g+2} = \frac{\frac{h(i)g+x_2}{g+1}(g+1)+x_1}{g+2}$$

Where x_1 only depends on b_1, and x_2 only depends on b_2. $\square$

Corollary 12 *Let A and B be sequences of observations from the same set of observations. Let Z^A,Z^B be states derived from these observations. Now, let Z^C be the state resulting from first incorporating A and later on B. Under these conditions it holds that:*

$$Z^C = Z^A \cup Z^B$$

Proof: This can be shown by induction on g using Lemma 11. □

In Corollary 12 no statement is made about the nature of the observations. So, both observers may have observed exactly the same actions, i.e. such equal observations are essentially one observation. If an observation b is incorporated multiple times, the relative frequencies $h(i)$ no longer represent the observed reality, as observation b is overvalued. So, if information from different sources is incorporated, duplicated information needs to be detected and incorporated only once.

5.4.3 Unlinkability

Steinbrecher and Köpsell [SK03b] propose another generalization which introduces the notion of unlinkability and takes also local anonymity into account. Particularly, they point out how anonymity can be quantified in terms of unlinkability. In contrast to anonymity which is a property of subjects, unlinkability could be a property between arbitrary items, that is subjects, actions, events, etc. Unlinkability of items with respect to an observation of the adversary holds, if the items are not more and not less related for the adversary before and after her observation. Thus, if a sender was anonymous before an adversary's observation and unlinkability holds with respect to the sender and her message, then the sender remains anonymous after the observation.

Steinbrecher and Köpsell model the relation between items as equivalence relation on the set of items. The adversary is supposed to know the set of items, but not to know the equivalence relation. However, the adversary may eventually gain knowledge about the equivalence relation. This gain in knowledge is modelled by a change in the probabilities for each possible equivalence relation.

This approach is first applied to model unlinkability between two items

and then successively to more complex issues. We denote the relation between two items a_i and a_j within a set[6] $\mathcal{A}$ as $a_i \sim_{r(\mathcal{A})} a_j$. Furthermore, we denote the probability which the adversary assigns to this relation as $\Pr(X = (a_i \sim_{r(\mathcal{A})} a_j))$ for a random variable X or, in short, $\Pr(a_i \sim_{r(\mathcal{A})} a_j)$. Accordingly, $\Pr(a_i \nsim_{r(\mathcal{A})} a_j)$ denotes the probability that the items a_i and a_j are not in relation.

The entropy $H(i,j) := H(X)$ can then be used as a measure for the degree of unlinkability $d(i,j)$ of the two items a_i and a_j.

$$\begin{aligned} d(i,j) &= H(i,j) \\ &= -\Pr(a_i \sim_{r(\mathcal{A})} a_j) \cdot \log_2\big(\Pr(a_i \sim_{r(\mathcal{A})} a_j)\big) \\ &\quad -\Pr(a_i \nsim_{r(\mathcal{A})} a_j) \cdot \log_2\big(\Pr(a_i \nsim_{r(\mathcal{A})} a_j)\big) \end{aligned} \tag{5.18}$$

The degree $d(i,j)$ becomes 0, if the adversary is either certain of $a_i \sim_{r(\mathcal{A})} a_j$ or of $a_i \nsim_{r(\mathcal{A})} a_j$. The degree becomes 1, if the adversary is completely uncertain of the relation between a_i and a_j, that is $\Pr(a_i \sim_{r(\mathcal{A})} a_j) = 0.5$ as well as $\Pr(a_i \sim_{r(\mathcal{A})} a_j) = 0.5$. The former case describes perfect linkability, whereas the latter case describes perfect unlinkability.

Similarly, the unlinkability of a set of items can be quantified. Let $A \subseteq \mathcal{A}$ be a subset of all items with $|A| > 2$ and $\sim_{r(A)}$ be an equivalence relation on A. The item set A denotes all items which an adversary observes and $\sim_{r(A)}$ denotes a guess of equivalence classes in A. Therefore, the probability for an adversary to succeed with linking is the probability for $\sim_{r(A)}$ being the same as $\sim_{r(\mathcal{A})}$, the actual equivalence relation, restricted to the elements of A, formally

$$\Pr\Big(\sim_{r(A)} = \sim_{r(\mathcal{A})}\big|_A\Big) \tag{5.19}$$

The degree of unlinkability with respect to A can then be calculated by means of the enumeration I_k of all possible equivalence relations on A and

[6] This could be an anonymity set. However, if we would explicitly write about anonymity sets here, we would unnecessarily lose generality for the types of items and entirely stick to subjects, instead.

the entropy of the corresponding probability distribution.

$$\begin{aligned} d(A) &= H(A) \\ &= -\sum_{j \in I_k} \frac{1}{|I_k|} p_j \cdot \log_2 p_j \\ &\quad \text{where } p_j = \Pr\Big(\sim_{r_j(A)} = \sim_{r(\mathcal{A})}\big|_A\Big) \end{aligned} \tag{5.20}$$

The degree $d(A)$ is 1, if the adversary is certain of one $\sim_{r_j(A)}$ being the same as $\sim_{r(\mathcal{A})}$ restricted to elements of A. The degree $d(A)$ is 0, if the adversary is completely uncertain about all $\sim_{r_j(A)}$, that is $\Pr(\sim_{r_j(A)} = \sim_{r(\mathcal{A})}|_A) = 0.5$ for each $r_j \in I_k$.

Steinbrecher and Köpsell pointed out, however, that it is not sufficient to address unlinkability within one set only. In order to describe anonymity in terms of unlinkability, it is rather necessary to address unlinkability between different sets. This could be, for instance, a set of messages and a set of senders. Sender anonymity can then be described by unlinkability between senders and messages.

The definition of unlinkability between two different sets $\mathcal{A}$ and $\mathcal{B}$ is similar to unlinkability between two items within the same set. The equivalence relation between two items within the same set, however, has to be replaced by a relation $\sim_{r(\mathcal{A},\mathcal{B})}$ between items in $\mathcal{A}$ and $\mathcal{B}$. The relation $\sim_{r(\mathcal{A},\mathcal{B})}$ itself is no equivalence relation, however, equivalence relations $\sim_{r(\mathcal{A})}$ and $\sim_{r(\mathcal{B})}$ can be constructed by means of capturing all $a \in \mathcal{A}$ in equivalence classes of $\sim_{r(\mathcal{A})}$ which are related to the same $b \in \mathcal{B}$ (and vice versa for $\sim_{r(\mathcal{B})}$).

The degree of linkability of two items within different sets $d(a, b)$ can then also be reduced to entropy.

$$\begin{aligned} d(a,b) &= H(a,b) \\ &= -\Pr(a \sim_{r(\mathcal{A},\mathcal{B})} b) \cdot \log_2\big(\Pr(a \sim_{r(\mathcal{A},\mathcal{B})} b)\big) \\ &\quad -\Pr(a \nsim_{r(\mathcal{A},\mathcal{B})} b) \cdot \log_2\big(\Pr(a \nsim_{r(\mathcal{A},\mathcal{B})} b)\big) \end{aligned} \tag{5.21}$$

5.4.4 Rényi Entropy

This section deals with phase [CAL], i.e. calculation of privacy parameters based on a given observer state. Thereby Sections 5.4.4.1 and 5.4.4.2 refer

to the case that a user has exactly one digital identity. Section 5.4.4.3 describes how calculations have to be done in case users may have multiple digital identities.

5.4.4.1 Quantifying Anonymity

Shannon entropy [Sha48] is often used as a metrics for anonymity. Given an observer state Z, the Shannon entropy $H_\varnothing$ of information b can be computed.

Definition 13 (Shannon entropy) *Let b be an observation and $\mathcal{V}_b$ a set of suspects related to observation b. The* Shannon entropy *of b in a state Z is the Shannon entropy of the suspects $\mathcal{V}_b$.*

$$H_\varnothing = -\sum_{(v \in \mathcal{V}_b)} p(v|b) \log_2 p(v|b) \tag{5.22}$$

$$p(v|b) = \frac{p(v \wedge (\bigvee_{(w \in \mathcal{V}_b)} w))}{p(\bigvee_{(w \in \mathcal{V}_b)} w)} \tag{5.23}$$

$$= \frac{h(v)}{\sum_{(i \in \mathcal{V}_b)} h(i)} \tag{5.24}$$

In (5.24), $h(i)$ denotes the probability of the identity i within the observer state Z.

Given a Shannon entropy $H_\varnothing$, $|\mathcal{S}| = 2^{H_\varnothing}$ denotes the equivalent size of a uniformly distributed anonymity set $\mathcal{S}$.

Example: The Shannon entropy of the observation $b = (0, 0, \perp)$ in Figure 5.1 is one Bit. This means that the suspects are as anonymous as they would be within a uniform distributed anonymity set of size two. $\diamondsuit$

The Shannon entropy $H_\varnothing$ specifies the average amount of information needed in addition to b in order to uniquely identify a digital identity.

Here we refer to the case that a user has only one digital identity, so that a measurement related to a digital identity can be seen synonymous to a measurement related to the user, who "owns" this digital identity. The Shannon entropy $H_\varnothing$ specifies the average amount of information needed in addition to b in order to uniquely identify a digital identity. In case of a

user evaluating her anonymity, she usually knows her digital identity. So, it may be more useful for her to compute the amount of information needed to identify *her*, i.e., her digital identity. This so called *individual anonymity* can be computed as follows:

$$H(i) = -\log_2 p(i|b) \tag{5.25}$$

From the viewpoint of each single user, *individual anonymity* is the most accurate anonymity measure.

Example: An observer knows that within a given source of information the element A shows up with a probability of 0.5 . If the observer is only interested in the occurrence of A (i.e. how anonymous A is), this is independent of the Shannon entropy of the information source. The anonymity measure of A only depends on the probability of A's occurrence. On the other hand, the Shannon entropy also depends on the number of elements of the information source. So, even if A occurs with a probability of 0.5, the Shannon entropy can have an arbitrarily high value depending on number and distribution of the other elements of the information source. But the amount of information needed to identify A remains — independent of the Shannon entropy of the information source — the same. ♢

It is also possible to specify a worst case measure for anonymity [THV04b]. This is the individual anonymity of the identity with the highest probability (also called *Min-entropy*):

$$H_{\text{Min}} = -\log_2 \max_{I_b}(p(i|b)) \tag{5.26}$$

In [CS06] Stefan Schiffner and I discussed usage of *Rényi entropy* as a more general metric for anonymity. Rényi entropy H_α, introduced by Rényi [Ren60], is defined as follows:

$$H_\alpha = \frac{1}{1-\alpha} \log_2 \sum_{(v \in \mathcal{V}_b)} p(v|b)^\alpha \tag{5.27}$$

Besides the probability distribution given, Rényi entropy incorporates an additional parameter α. In Figure 5.3 the influence of α on Rényi entropy is shown. The more α grows the more the Rényi entropy converges to Min-entropy H_{Min}. On the other hand, the more α runs to zero the more

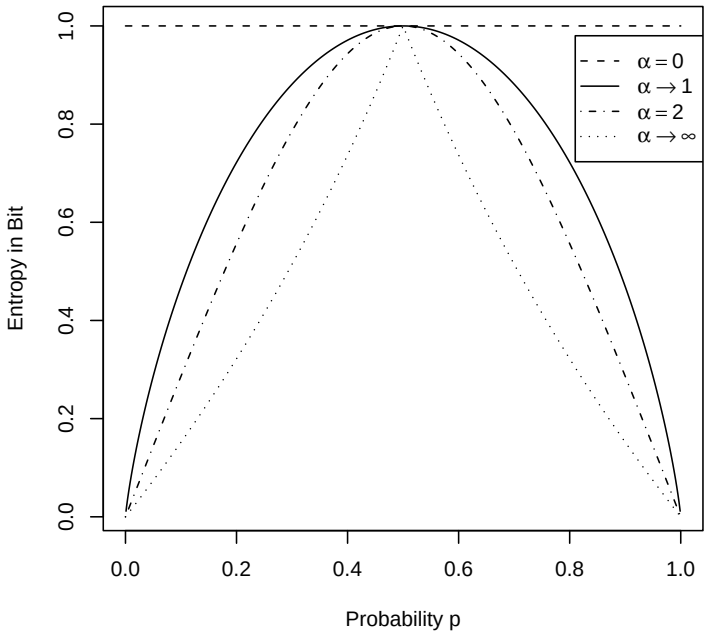


Figure 5.3: Influence of parameter α on Rényi entropy of a source containing two elements with probabilities p and $1 - p$ resp.

H_α converges to Max-entropy $H_{\text{Max}} = \log_2 N$, where N is the number of elements of the probability distribution given[7]. Furthermore, if α runs to 1, Rényi entropy converges to Shannon entropy. The proofs of these facts are given in [CS06].

By adjusting the parameter α, it is possible to fade between worst-case anonymity, average case anonymity, and k-anonymity. So, for evaluating anonymity within a system given, the parameter α can be adapted according to certain characteristics of the system.

[7] H_{Max} directly corresponds to k-anonymity. It denotes the entropy of a source with k elements, thereby ignoring the probability distribution of the elements, i.e., assuming a uniform probability distribution.

5.4.4.2 Quantifying Linkability of Actions

Regarding linkability, it is interesting for a user, to what extent it can be determined that actions have been done by the same user. More formally, there are two actions c_1 and c_2 which have been observed in the form of observations b_1 and b_2.

According to [SK03a], linkability of items of interest can be measured regarding equivalence classes, for which (after observations) an attacker has partial knowledge about which items of interest belong to which class.

Applied to the model used here, the equivalence classes are the digital identities. By an observation of an action, suspect digital identities can be determined corresponding to the observation of this action (see Definition 3), i.e. information about association of items of interest (actions) to equivalence classes (digital identities) is gained.

Regarding observations b_1 and b_2, the suspect sets are $\mathcal{V}_{b_1}$ resp. $\mathcal{V}_{b_2}$. Within a set of suspects, a digital identity has the probability $p(v|b)$, which is derived from the current observer state as shown in equations (5.23) and (5.24).

The probability p_r, that actions c_1 and c_2 belong to the same digital identities, can be computed as follows:

$$p_r = \sum_{(v \in \mathcal{V}_{b_1 \wedge b_2})} p(v|b_1) \cdot p(v|b_2)$$

Thereby, $\mathcal{V}_{b_1 \wedge b_2}$ denotes the set of digital identities, which are contained in both sets $\mathcal{V}_{b_1}$ and $\mathcal{V}_{b_2}$, i.e. which are suspects of both observations, b_1 and b_2. According to [SK03a], the probability $p_{\neg r}$, that the actions c_1 and c_2 do not belong to the same digital identity is $1 - p_r$.

From probabilities p_r and $p_{\neg r}$ a degree of linkability d can be computed by using the Shannon entropy [SK03a]:

$$d := H(p_r, p_{\neg r}) = -p_r \cdot \log_2 p_r - p_{\neg r} \cdot \log_2 p_{\neg r}$$

The events "actions c_1 and c_2 belong to the same digital identity" and "actions c_1 and c_2 do not belong to the same digital identity" are used as elements of a two-element source of information. The degree of linkability d is the Shannon entropy of this source. It specifies, how much an observer has

learnt about the relation between c_1 and c_2 from observations $\mathcal{V}_{b_1}$ and $\mathcal{V}_{b_2}$, taking also into account the a-priori knowledge about the digital identities derived from the current observer state.

The maximum degree of linkability, $d = 1$, means that the observer does not know anything about whether actions c_1 and c_2 belong to the same digital identity or not.

If $p_r > p_{\neg r}$, the degree denotes the certainty of the observer, that actions c_1 and c_2 *belong to the same digital identity*, otherwise it denotes the certainty of the observer that the actions do *not belong to the same digital identity*.

In case a user has only one digital identity, linkability related to a digital identity is the same as linkability related to a user. The next section deals with users having multiple digital identities.

5.4.4.3 Users with Multiple Digital Identities

In real life, a user will often not only have one digital identity, but lots of them. So, for example a user may have many different e-mail addresses, which she uses in different situations. Nevertheless also in this case, a user will be interested in *her* privacy, and not only in the privacy of one of her digital identities.

In order to calculate privacy parameters for users having multiple identities, we can first determine suspect digital identities as described for the different metrics in the previous sections. Now, in order to calculate measurements with respect to *users*, we need to group suspect digital identities belonging to the same user into *personal* digital identities. Thereby, grouping means that for each user the probability values of all digital identities belonging to this user are summed up. For a meaningful per-user-grouping of digital identities the observer state used must contain attributes, which can be used to distinguish between different users. If this is not the case, such an observer state cannot be used to determine privacy parameters of *users*.

After the probabilities of *personal* digital identities are determined, calculations of anonymity and linkability metrics can be done as described above, but based on probabilities of these *personal* digital identities.

Chapter 6

Conclusions

This document, in a close relation to the deliverable D13.1 *Identity and impact of privacy enhancing technologies*, provided not only a critical review of issues relevant to identity and privacy modelling, but also some insight into the potential of existing relevant privacy modelling approaches.

Majority of models that we have identified in this and earlier research rely in one way or another on *context information* that describes entity behaviour in a system. Privacy/content models discussed in this deliverable aim to process the data to learn frequent behavioral patterns as well as to decide how sensitive information this data may contain. Metrics for privacy quantification then aim to measure the degree of privacy a protocol or a communication system can provide to its users. Another purpose of such quantification is to provide a metric for the (level of) privacy which users actually may expect with respect to their situation, e.g., considering their previous actions.

We reviewed most promising context information models and behaviour modelling approaches, starting with context information models like the Freiburg Privacy Diamond and PATS, but also models based on set theory, directed graphs, first-order logic and finally a variant of Object-Role modelling. With respect to the behaviour modelling techniques, we discussed the global mixture model, maximum entropy model and hidden Markov model, together with a discussion of data mining (namely cluster analysis) and its importance to user behaviour modelling.

Finally, we analyzed existing metrics for privacy properties quantification (anonymity, pseudonymity, unlinkability, unobservability), starting with formal methods that have been proposed for determination of the degree of anonymity, namely "function views", but also addressing privacy in (statistical) databases with respect to possible risk of entities' re-identification and the degree of user's anonymity provided in systems like the DC net, mix-based systems or Crowds.

Deliverable D13.1 and this deliverable 13.6 are now essential for the follow-up work of FIDIS WP13, where the plan is as follows:

- *Deliverable D13.8: Applicability of privacy models*, where we plan to use some privacy modelling approaches in use-cases involving profiling, systems using different forms of identities, etc. The goal of that deliverable will be to review/illustrate the applicability of models from this deliverable D13.6, and this deliverable actually went few steps ahead in this way.
- *Deliverable D13.9: Estimating quality of identities* that will extend our previous 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.

Bibliography

[BBA+00] Matthias Baumgarten, Alex G. Büchner, Sarabjot S. Anand, Maurice D. Mulvenna, and John G. Hughes. User-Driven Navigation Pattern Discovery from Internet Data. In *Web Usage Analysis and User Profiling: International WEBKDD99 Workshop San Diego, CA, USA, August 15, 1999*, volume 1836. Springer-Verlag, 2000.

[Bel97] M. Bellare. A note on negligible functions. Technical Report CS97-529, Department of Computer Science and Engineering, UCSD, 1997.

[Ber02] Pavel Berkhin. Survey Of Clustering Data Mining Techniques. Technical report, Accrue Software, San Jose, CA, USA, 2002.

[BKP90] J. G. Bethlehem, W. J. Keller, and J. Pannekoek. Disclosure control of microdata. *Journal of The American Statistical Association*, 85:38–45, 1990.

[BL00] José Borges and Mark Levene. Data Mining of User Navigation Patterns. In *Web Usage Analysis and User Profiling: International WEBKDD99 Workshop San Diego, CA, USA, August 15, 1999*, volume 1836. Springer-Verlag, 2000.

[Boa05] Common Criteria Editorial Board. *Common Criteria for Information Technology Security Evaluation (Part 2: Security functional requirements)*, version 2.3, August 2005.

[Bro00] Alan J. Broder. Data Mining, the Internet, and Privacy. In *Web Usage Analysis and User Profiling: International WEBKDD99 Workshop San Diego, CA, USA, August 15, 1999*, volume 1836. Springer-Verlag, 2000.

[CHM+00] Igor Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Visualization of Navigation Patterns on a Web Site Using Model-based Clustering. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–284, Boston, Massachusetts, United States, 2000. ACM Press.

[CKM05] Daniel Cvrček, Marek Kumpošt, and Václav Matyáš. On Privacy Classification in Ubiquitous Computing Systems. *International Scientific Journal of Computing*, 4(2):8, 2005.

[CKM06] Daniel Cvrček, Marek Kumpošt, and Václav Matyáš. A Privacy Classification Model Based on Linkability Valuation. In *Security and Embedded Systems*, pages 91–98. Kluwer Academic Publishers, 2006.

[CM96] Chris Clifton and Don Marks. Security and Privacy Implications of Data Mining. In *Workshop on Data Mining and Knowledge Discovery*, Montreal, Canada, 1996. University of British Columbia Department of Computer Science.

[CS06] Sebastian Clauß and Stefan Schiffner. Structuring anonymity metrics. In Atsuhiro Goto, editor, *Proceedings of the second ACM workshop on Digital identity management*, pages 55–62, Alexandria, Virginia, USA, November 2006. ACM.

[Dan03] G. Danezis. *Better Anonymous Communications.* PhD thesis, University of Cambridge, December 2003.

[DIC00] Sumeet Dua, S. S. Iyengar, and Eungchun Cho. Discovery of Web Frequent Patterns and User Characteristics from Web Access Logs: A Framework for Dynamic Web Personalization. In *Proceedings of the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET'00)*, Washington, DC, USA, 2000. IEEE Computer Society.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.

[DSCP02] C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *Workshop on Privacy Enhancing Technologies*, volume 2482 of *LNCS*, 2002.

[ESM02a] Yuval Elovici, Bracha Shapira, and Adlai Maschiach. A New Privacy Model for Hiding Group Interests while Accessing the Web. In *WPES '02: Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, pages 63–70, New York, NY, USA, 2002. ACM Press.

[ESM02b] Yuval Elovici, Bracha Shapira, and Adlai Maschiach. New Privacy Model for Web Surfing. In A. Halevy and A. Gal, editors, *Next Generation Information Technologies and Systems : 5th International Workshop, NGITS 2002, Caesarea, Israel, June 24-25, 2002*, volume 2382. Springer-Verlag, 2002.

[FH87] Simone Fischer-Hübner. Zur reidentifikationssicheren statistischen Auswertung personenbezogener Daten in staatlichen Datenbanken. Diplomarbeit, Universität Hamburg, December 1987. In German.

[FH01] Simone Fischer-Hübner. *It-security and privacy: Design and use of privacy-enhancing security mechanisms*, volume 1958 of *Lecture Notes in Computer Science*. Springer, 2001.

[FSS00] Yongjian Fu, Kanwalpreet Sandhu, and Ming-Yi Shih. A Generalization-Based Approach to Clustering of Web Usage Sessions. In *Web Usage Analysis and User Profiling: International WEBKDD99 Workshop San Diego, CA, USA, August 15, 1999*, volume 1836. Springer-Verlag, 2000.

[HDSB03] Younes Hafri, Chabane Djeraba, Peter Stanchev, and Bruno Bachimont. A Web User Profiling Approach. In X. Zhou, Y. Zhang, and M.E. Orlowska, editors, *Web Technologies and Applications: 5th Asia-Pacific Web Conference, APWeb 2003, Xian, China, April 23-25, 2003*, volume 2642. Springer-Verlag, 2003.

[HIM05] Karen Henricksen, Jadwiga Indulska, and Ted McFadden. Modelling Context Information with ORM. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems 2005: OTM Workshops: OTM Confederated Internationl Workshops and Posters, AWeSOMe, CAMS, GADA, MIOS+INTEROP, ORM, PhDS, SeBGIS, SWWS, and WOSE 2005, Agia Napa, Cyprus, October 31 - November 4, 2005*, volume 3762. Springer-Verlag, 2005.

[HO03] J. Halpern and K. O'Neill. Anonymity and information hiding in multiagent systems, 2003.

[HO05] Joseph Y. Halpern and Kevin R. O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 13(3):483–514, May 2005.

[HS04a] D. Hughes and V. Shmatikov. Information hiding, anonymity and privacy: A modular approach. *Journal of Computer Security, special issue on selected papers of WITS 2002*, 12(1):3–36, 2004.

[HS04b] D. Hughes and V. Shmatikov. Information hiding, anonymity and privacy: a modular approach, 2004.

[HWMI05] Karen Henricksen, Ryan Wishart, Ted McFadden, and Jadwiga Indulska. Extending Context Models for Privacy in Pervasive Computing Environments. In *3rd IEEE Conference on Pervasive Computing and Communications Workshops (PerCom 2005 Workshops)*, Kauai Island, HI, USA, 2005. IEEE Computer Society.

[JW02] A. R. Johnson and W. D. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, Upper Saddle River, New Jersey, 4th edition, 2002.

[Lek00] George Konstantinou Lekeas. *Data Mining the Web: The Case of City Universitys Log Files*. PhD thesis, City University London, 2000.

[Mal02] Bradley Malin. Compromising privacy with trail reidentification: The reidit algorithms. Technical Report CMU-CALD-02-108, Carnegie Mellon University, 2002.

[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.

[MPG03] Eren Manavoglu, Dmitry Pavlov, and C. Lee Giles. Probabilistic User Behavior Models. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)*, Washington, DC, USA, 2003. IEEE Computer Society.

[PH01] Andreas Pfitzmann and Marit Hansen. Anonymity, unobservability, and pseudonymity: A proposal for terminology. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies (PET'00)*, volume 2009 of *LNCS*, pages 1–9. Springer-Verlag, 2001.

[RCRM02] Anand Ranganathan, Roy T. Campbell, Arathi Ravi, and Anupama Mahajan. ConChat: A Context-Aware Chat Program. *IEEE Pervasive Computing*, 1(3):51–57, July–September 2002.

[RDN05] M.A. Razzaque, Simon Dobson, and Paddy Nixon. Categorization and Modelling of Quality in Context Information. In *Proceedings of the IJCAI 2005 Workshop on AI and Autonomic Communications*, 2005.

[Ren60] Alfred Renyi. On measures of entropy and information. In *Fourth Berkeley Symposium Math. Statist. and Prob.*, pages 547–561, Berkeley, 1960.

[RI00a] K. Rannenberg and G. Iachello. Protection profiles for remailer mixes. In *International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability*, volume 2009 of *LNCS*, pages 181–230, Berkeley, California, 2000. Springer-Verlag.

[RI00b] K. Rannenberg and G. Iachello. Protection profiles for remailer mixes – do the new evaluation criteria help? In *16th Annual Computer Security Applications Conference (ACSAC'00)*, pages 107–118. IEEE Computer Society, December 2000.

[RR98] Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[Rub93] D. B. Rubin. Discussion: Statistical disclosure limitation. *Journal of Official Statistics*, 9:462–468, 1993.

[SAT+99] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced Interaction in Context. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 89–101, London, UK, 1999. Springer-Verlag.

[SD02a] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity, 2002.

[SD02b] Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul Syverson, editors, *Designing Privacy Enhancing Technologies (PET'02)*, volume 2482 of *LNCS*, pages 41–53. Springer-Verlag, 2002.

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

[Shm04] V. Shmatikov. Probabilistic model checking of an anonymity system, 2004.

[SK03a] Sandra Steinbrecher and Stefan Köpsell. Modelling unlinkability. In Roger Dingledine, editor, *Designing Privacy Enhancing Technologies (PET'03)*, volume 2760 of *LNCS*, pages 32–47. Springer-Verlag, 2003.

[SK03b] Sandra Steinbrecher and Stefan Köpsell. Modelling unlinkability. In Roger Dingledine, editor, *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*, pages 32–47. Springer-Verlag, LNCS 2760, March 2003.

[Sol06] Daniel Solove. A taxonomy of privacy. *University of Pennsylvania Law Review*, 154(3):477–560, 2006.

[SS96] Steve Schneider and Abraham Sidiropoulos. CSP and anonymity. In *ESORICS*, pages 198–218, 1996.

[SS98] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, SRI Computer Science Laboratory, 1998.

[SS99] Paul F. Syverson and Stuart G. Stubblebine. Group principals and the formalization of anonymity. In *World Congress on Formal Methods (1)*, pages 814–833, 1999.

[Swe02a] Latanya Sweeney. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[Swe02b] Latanya Sweeney. k-Anonymity: A model for protecting privacy. *International Journal on Uncertainity, Fuzziness and Knowledge-based Systems*, 10(5):557–570, May 2002.

[Tav99] Herman T. Tavani. Information Privacy, Data Mining, and the Internet. In *Ethics and Information Technology*, Hingham, MA, USA, 1999. Kluwer Academic Publishers.

[The99] The Common Criteria Project Sponsoring Organisations. *Common Criteria for Information Technology Security Evaluation – part 2, version 2.1.* August 1999.

[THV04a] G. Toth, Z. Hornak, and F. Vajda. Measuring anonymity revisited, 2004.

[THV04b] Gergely Tóth, Zoltán Hornák, and Ferenc Vajda. Measuring anonymity revisited. In Sanna Liimatainen and Teemupekka Virtanen, editors, *Proceedings of the Ninth Nordic Workshop on Secure IT Systems*, pages 85–90, Espoo, Finland, November 2004.

[WHI05] Ryan Wishart, Karen Henricksen, and Jadwiga Indulska. Context Obfuscation for Privacy via Ontological Descriptions. In Thomas Strang and Claudia Linnhoff-Popien, editors, *Location- and Context-Awareness: First International Workshop, LoCA 2005, Oberpfaffenhofen, Germany, May 12-13, 2005*, volume 3479. Springer-Verlag, 2005.

[XZ01] Jitian Xiao and Yanchun Zhang. Clustering of Web Users Using Session-Based Similarity Measures. In *Proceedings of the 2001 International Conference on Computer Networks and Mobile Computing (ICCNMC'01)*, Washington, DC, USA, 2001. IEEE Computer Society.

[XZJL01] Jitian Xiao, Yanchun Zhang, Xiaohua Jia, and Tianzhu Li. Measuring Similarity of Interests for Clustering Web-users. In *Proceedings of the 12th Australasian conference on Database technologies*, Washington, DC, USA, 2001. IEEE Computer Society.

[YD02] Dit-Yan Yeung and Yuxin Ding. User Profiling for Intrusion Detection Using Dynamic and Static Behavioral Models. In M.-S. Chen, P.S. Yu, and B. Liu, editors, *6th Pacific-Asia Conference,*

PAKDD 2002, Taipel, Taiwan, May 6-8, 2002, volume 2336. Springer-Verlag, 2002.

[ZKM03a] A. Zugenmaier, M. Kreutzer, and G. Müller. The Freiburg Privacy Diamond: An attacker model for a mobile computing environment. In *Kommunikation in Verteilten Systemen (KiVS) '03*, Leipzig, 2003.

[ZKM03b] Alf Zugenmaier, Michael Kreutzer, and Günter Müller. The Freiburg Privacy Diamond: An Attacker Model for a Mobile Computing Environment. In *Kommunikation in Verteilten Systemen (KiVS) '03*, Leipzig, 2003.

[Zug03] A. Zugenmaier. *Anonymity for Users of Mobile Devices through Location Addressing.* RHOMBOS-Verlag, ISBN 3-930894-96-3, Berlin, 2003.