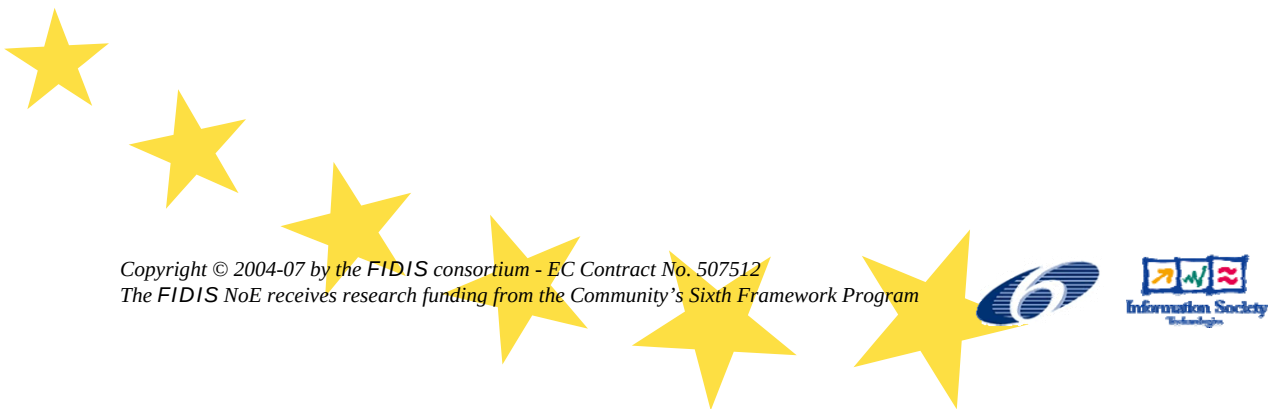
FIDIS

Future of Identity in the Information Society

Title:	"D13.1: Identity and impact of privacy enhancing technologie"
Author:	WP13
Editors:	Daniel Cvrček (MU, Czech Republic) Vashek Matyáš (MU, Czech Republic)
Reviewers:	Jozef Vyskoč (VaF, Slovakia)
Identifier:	D13.1
Type:	[Deliverable-Report]
Version:	1.0
Date:	Sunday, 13 May 2007
Status:	[Final]
Class:	[Public]
File:	fidis-wp13-del13.1.final.pdf

Brief Summary

This document is a report on technologies that enhance privacy from the technological point of view. We examined neither policy-based solutions nor law, we provide a review of technologies available.



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

Copyright Notice:

Members of the FIDIS consortium

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

Versions

Version	*Date*	*Description (Editor)*
0.1	12.02.2007	• Initial draft
0.2	11.03.2007	• Updated version of chapters • Minor formatting updates
0.7	2.4.2007	• Submitted for internal review
0.9	9.5.2007	• Revision after internal review
1.0	13.5.2007	• 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)
- Katholieke Universiteit Leuven, COSIC, Belgium (7)
- Technische Universität Dresden, Germany (11)

Management Summary

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

While the primary goal of this report is to provide na overview of computer technologies enhancing privacy/anonymity of users, this overview will also serve as a starting point for the research of theoretical foundations that support definitions of privacy needed for its modelling and measuring — work planned for the WorkPlan 4. This future work should provide an insight to a possibly critical ability for the future — to express the level of protection/state of identity related information.

Chapter two of this report presents the terminology and concepts crucial to this document. Chapter three reviews metrics that have been developed in order to assess designs providing privacy-preserving properties, such as anonymity and unlinkability, and metrics for anonymized data with respect to the k-anonymity model. The fourth chapter then reviews attacker models for existing systems, together with estimates of attack impacts. Since this work will be followed by research on privacy models, we have judged is as crucial to put together a solid synthetical overview of attacks and their importance. The fifth chapter overviews most common privacy primitives and the next chapter focuses on application privacy. Finally, chapter seven discusses communication privacy in detail. This chapter concludes with general principles in computer and network systems.

Contents

Chapter 1

Introduction

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

While the primary goal of this report is to provide na overview of computer technologies enhancing privacy/anonymity of users, this overview will also serve as a starting point for the research of theoretical foundations that support definitions of privacy needed for its modelling and measuring — work planned for the WorkPlan 4. This future work should provide an insight to a possibly critical ability for the future — to express the level of protection/state of identity related information.

Chapter two of this report presents the terminology and concepts crucial to this document. It starts with the terminology of Pfitzmann and Hansen [62], who started their work in 2000 with a set of working definitions for anonymity, unlinkability, unobservability, and pseudonymity. Since then these definitions have been subject to rigorous public discussions and the results then adopted in most of the anonymity literature. This part is followed by a brief review of the k-anonymity model for anonymizing personal records in a database as proposed by Samarati and Sweeney in [71, 79]. Common Criteria [81] bring yet another approach to describing privacy — while being a standard for security evaluations of IT products and systems, it also treats privacy as one of possible properties of such systems. Finally, we also incorporated the Freiburg Privacy Diamond approach of A. Zugenmaier et al. [86, 87] for mobile environments.

Chapter three reviews metrics that have been developed in order to assess designs providing privacy-preserving properties, such as anonymity and unlinkability, and metrics for anonymized data with respect to the k-anonymity model.

The fourth chapter then reviews attacker models for existing systems, together with estimates of attack impacts. Since this work will be followed by research on privacy models, we have judged is as crucial to put together a solid synthetical overview of attacks and their importance.

The fifth chapter overviews most common privacy primitives like encryption, secret sharing, commitment and zero-knowledge schemes, blind signatures, pseudonymous convertible credentials and pseudonyms, private information retrieval, and finally steganography.

The next chapter focuses on application privacy. We start with a review of privacy issues in databases — first reviewing relevant aspects of so-called statistical databases and then private database queries, together with privacy and data mining issues. The next part deals with privacy issues in e-commerce and namely in e-cash systems. Privacy in location-aware systems and relevant aspects of identity management are topics of following discussion.

Chapter seven discusses communication privacy in detail. It starts with simple-proxy systems, then it presents a well-known Crowds system [66], broadcast approach, RING Network, so-called buses, DC (Dining Cryptographers) networks, mixes, private information retrieval. This chapter concludes with general principles in computer and network systems.

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 [77] 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 3, we present the existing metrics for quantification of privacy properties.

2.2 Pfitzmann-Hansen Terminology

Pfitzmann and Hansen [62] 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) 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).

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 [71, 79]. 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 [56], 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.

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 [79, 71]: "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 [81] 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. Let us now introduce the concepts of the privacy definition as defined in Common Criteria documents. Let us briefly summarise relevant Common Criteria notions and concepts.

Target of Evaluation (TOE) – An IT product or system and its associated administrator and user guidance documentation that is the sub-

ject 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.

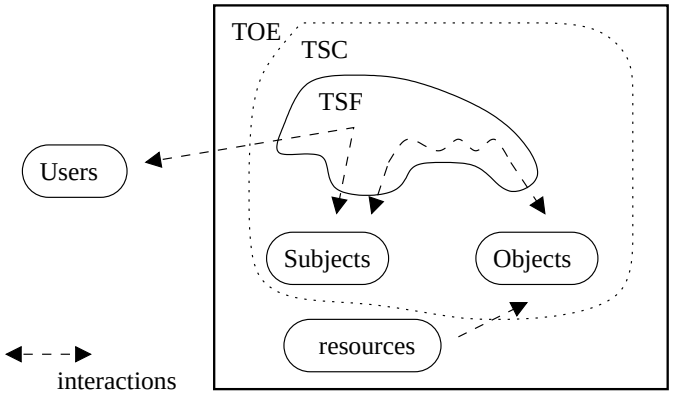


Figure 2.1: Common Criteria model.

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 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).

2.5 Freiburg Privacy Diamond

FPD is a semiformal anonymity (and partly also unlinkability) model by A. Zugenmaier et al. [86, 87]. 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. 2.2). The main reason for introducing *location* as a category here is probably due to the overall focus of this model on mobile computing.

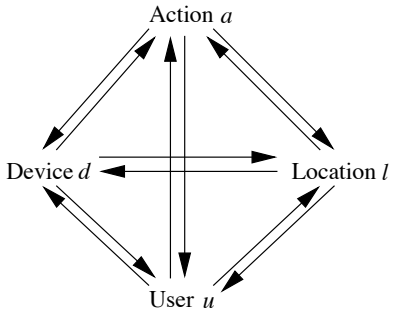


Figure 2.2: Freiburg Privacy Diamond.

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 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 [86] 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.

Chapter 3

Metrics

Metrics have been developed in order to assess designs providing privacy-preserving properties, such as anonymity and unlinkability. In this chapter, we introduce the anonymity and unlinkability metrics that have been proposed in the literature.

3.1 Metrics for Anonymous Communication

Most attacks on anonymous communication networks provide the adversary with probabilistic information on the identity of the entities communicating with each other. This is the reason why information theoretical anonymity metrics [32, 73] have been widely adopted to quantify the anonymity provided by a variety of designs.

But there had been some attempts to quantify anonymity in communication networks before information theoretic anonymity metrics were proposed.

Reiter and Rubin [66] define *degree of anonymity* as a probability $1 - p$, where p is the probability assigned by an attacker to potential senders. In this model, users are more anonymous as they appear (towards a certain adversary) to be less likely of having sent a message. This metric considers users separately, and therefore does not capture anonymity properties very well. Consider a first system with 2 users which appear to be the sender of a message with probability 1/2. Now consider a second system with 1000 users. User u_1 appears as the sender with probability 1/2, while all the other users are assigned probabilities of having sent the message below 0.001. According to the definition of Reiter and Rubin, the *degree of anonymity* of u_1 and of the two users of the first system would be the same (50%). However,

in the second system, u_1 looks much more likely to be the sender than any other user, while the two users of the first system are indistinguishable to the adversary.

Berthold *et al.* [8] define the *degree of anonymity* as $A = log_2(N)$, where N is the number of users of the system. This metric only depends on the number of users of the system, and therefore does not express the anonymity properties of different systems. The total number N of users may not be known. Moreover, adversaries may be able to obtain probabilistic information on the set of potential senders, which is not taken into account in this metric.

Information theoretic anonymity metrics were independently proposed in two papers presented at the *2nd Workshop on Privacy Enhancing Technologies*. The profound principle of both metrics is the same. The metric proposed by Serjantov and Danezis [73] uses entropy as measure of the *effective anonymity set size*. The metric proposed by Diaz *et al.* [32] goes one step further, normalizing the entropy to obtain a *degree of anonymity* in the scale 0..1.

The quantification of anonymity is dependent on the adversary considered. The adversary has certain capabilities and deploys attacks in order to gain information and to find links between subjects and items of interest. Most of these attacks lead to a distribution of probabilities that assign subjects a certain probability of being linked to the items of interest. In this respect, a clear and detailed formulation of the attack model considered is a necessary step to measure the anonymity provided with respect to that attacker.

The information theoretic concept of entropy [76] provides a measure of uncertainty of a random variable. Let X be the discrete random variable with probability mass function $p_i = Pr(X = i)$, where i represents each possible value that X may take with probability $p_i > 0$. In this case, each i corresponds to a subject of the anonymity set; i.e., p_i is the probability of subject i being linked to the item of interest.

The entropy thus describes the amount of information (measured in bits) contained in the probability distribution of links between a set of subjects (the anonymity set) and a subject of interest. In [73], the entropy is proposed as a measure of the effective anonymity set size. If the entropy is normalized by the maximum the system could provide (if it was perfect and leaked no information) for a given number of users, we obtain a degree of anonymity [32] that gives a performance measure of anonymity provider.

The above mentioned approaches [32, 73] use Shannon-Entropy for measuring anonymity. Tóth et al. show in [82] that depending on the goal of

anonymity quantification, a worst case metric for anonymity is preferable over Shannon-Entropy, as the latter measures average case. They call this worst case metric *local* anonymity. It describes the anonymity of the most likely element within a probability distribution given.

In [23] Clauß and Schiffner propose Rényi-Entropy [68] as a more general information theoretic metric for anonymity. Besides the probability distribution describing the links between a set of subjects and an item of interest, Rényi-Entropy bases on an additional parameter α, which can be used to fade between worst-case anonymity, average case anonymity and k-anonymity. So, for evaluating anonymity within a given system, a parameter α can be adapted according to certain characteristics of the system.

A shortcoming of entropy based measurements (except the worst-case measurement) is, that the influence of outliers is strong (i.e., if some element of the probability distribution is very unlikely, the entropy of the source increases very much). This has the effect that even if the entropy of a source may be very high, this may not say much about anonymity of specific elements of this source. For overcoming this, Clauß and Schiffner [23] propose quantiles, which can be used to cut of lower bound outliers. So, quantiles can determine a worst case metric for the remaining elements (excluding the outliers).

3.2 Metrics for Unlinkability

A formalisation of relative anonymity is given in [78]. Beginning with a simple system model for unlinkability within one set a model for unlinkability between sets which tries to meet real world conditions slightly better is given. This model is based on information theory and describes unlinkability with equivalence classes. A special focus is on the possible attacker on unlinkability: If an attacker only learns the numbers of linkable items within a set his a posteriori probabilities of unlinkability will have increased in comparison to his a priori probabilities.

Hughes and Shmatikov develop a modular approach to specify unlinkability and other privacy properties based on a mathematical abstraction describing the partial knowledge of a function (the so-called function view [50]). In contrast to [78], equivalence relations are used to describe an attacker's inability to distinguish between system configurations (observational equivalence). This approach can be used independently of the underlying algebra or logic. But unfortunately in this approach probabilism is not included that makes it unsuitable as unlinkability metrics.

3.3 Metrics for Anonymized Data

In order to anonymize the data records contained in a database, it is not enough to remove obvious identifiers such as name, national ID number, or address. Often, if no further precautions are taken when releasing the data, the remaining attributes may constitute an identifier, or quasi-identifier, that allows for re-identification of the data subject when combined with other sources of information (e.g. the census).

The k-anonymity model [71, 79], introduced in Section 2.3, proposes a method to generalize attribute values, such that each possible query consisting of a combination of attribute values gives as result a set of at least k records.

Therefore, the anonymity set size for each query result is k. From the perspective of information-theoretic anonymity metrics, we can consider that the probability of matching a particular individual to the results of the query is uniformly distributed over the k resulting records.

Chapter 4

Attacker Models for Anonymity Systems

It is very hard to define a realistic attacker model for real-world systems. Internet communication does not respect any physical boundaries like country borders and one cannot forsee strength of possible attackers. The attacker may be an individual, an ISP, someone who owns several ISPs that keep their original brand names, and it can also be a state agency.

The good thing for anonymit is that it is very hard to predict how any communication session will be routed between its end points. The Internet is quite decentralised and self-regulated. When a part of the network drops out the data will find a different route to its destination. At the same time, the very same feature can be used to change routing metrics for a purpose of malicions attacks. This can be carried out with only restricted resources.

Almost every paper and article dealing with anonymous communication defined or at least imlicitly assumed certain types of attackers. We give a list of definitions from several sources before we define our own hierarchy that is compiled from all these sources.

4.1 Definitions of Attackers By Others

D. Chaum: Untraceable Electronic Mail, Return addresses, and Digital Pseudonyms [16]

One of the oldest papers and definitely the one most often cited. This paper defines the basic types of adversaries that are used till today. The classification is based partially on Dolev-Yao model. There are two definitions related to attackers in the paper.

1. No one can determine anything about the correspondences between a set of sealed items and the corresponding set of unsealed items, or create forgeries without the appropriate random string or private key.

2. Anyone may learn the origin, destination(s), and representation of all messages in the underlying telecommunication system and anyone may inject, remove, or modify messages.

Andreas Pfitzmann, Michael Waidner: Networks Without User Observability [64]

The authors here argue that eavesdropping can be eliminated. "Eavesdropping can be foiled by link-by-link encryption, but this does not foil attackers at the stations (e.g. via Trojan Horses)."

They further elaborate on several possible attackers: the administration, foreign states, companies, one's neighbours and communication partners. During the design of an anonymous network these possible attackers have to be translated into terms of stations and lines. A station is always under control of its owner and might be under control of everybody who has had access to it so far, e.g. its manufacturer, because he might have installed a Trojan Horse. Trojan Horses are a serious problem in stations with high complexity, e.g. switching centers. In simple user stations they can be detected (if tried) more easily. Lines are assumed to be owned by the PTT. Normally they can easily be observed by the PTT or an eavesdropper, but by physical measures such an attack can be made much more difficult.

They also define two different security goals: strong and weak. *Given a model of the attacker we have to define what we want to keep hidden from him. A strong possibility is to keep the sender and the recipient of a message secret. A weaker possibility is to keep only their relationship secret, i.e. sending and receiving of physical messages is observable, but it is infeasable for an attacker to link the physical message sent by the sender and the physical message received by the recipient.*

Michael Waidner and Birgit Pfitzmann: The dining cryptographers in the disco:unconditional sender and recipient untraceability with computationally secure servicability [83]

If unconditional untraceability of the senders of messages is is to be satisfied, the attacker may be computationally unlimited, able to eavesdrop communication between any two of the participants, and control an arbitrary subset A of the set of participants P (although at least two honest participants must exist if untraceability is to make sense).

ISDN-MIXes: Untraceable Communication with Very Small Bandwidth Overhead [63]

Attacker model is defined by possible attacks. There are three different classes of attacks mentioned in this paper.

1. *Attack on A as a sender*: If MIX_1, or an active attacker on A's subscriber line, changes the content of A's channel and colludes with B, they can test whether A is B's current communication partner, because B will receive nonsense in this case.

2. *Attack on the broadcast of incoming-call messages*: A second attack is to disturb the broadcast of incoming-call messages: Assume LA, or an active attacker on the subscriber lines, causes an incoming-call message to be handed to A only. Then, if the call is accepted, A must be the recipient.

3. *Blocking A's resources*: An attacker may, e.g., establish two calls with A. If he can then establish a call with X, too, then X $\neq$ A. This kind of attack cannot be prevented. However, it needs the collusion of the communication partner of X again; and as long as A's ressources are not fully used, she knows that she is not a victim of such an attack.

We can define the attackers as active local, or adaptive active local – depending on whether A is set in advance or during the attack.

David M. Goldschlag, Michael G. Reed, and Paul F. Syverson: Hiding Routing Information [46]

This paper assumes two types of attacks. The first one consists of possible traffic analysis attacks. The second type of attacks assume compromise of some of the routing nodes.

Kesdogan, Egner, Buschkes: Stop-and-Go MIXes: Providing Probabilistic Anonymity in an Open System [54]

Attacker model is equivalent to the one from D. Chaum in [18]. It assumes omnipresent attacker able to eavesdrop all communication lines and control all but one switching nodes. The attacker cannot, however, breat cryptographic security.

Michael Reiter and Aviel Rubin: Crowds: Anonymity for Web Transactions [66]

The introduced system is trying to protect against three different types of attackers:

1. A local eavesdropper is an attacker who can observe all (and only) communication to and from the users computer.

2. Collaborating crowd members are other crowd members that can pool their information and even deviate from the prescribed protocol.

3. The end server is the web server to which the web transaction is directed.

However, there are some limitations imposed on the attacks. For example, collaborating members and the end server cannot eavesdrop on communication between other members. Similarly, a local eavesdropper cannot eavesdrop on messages other than those sent or received by the users computer. A local eavesdropper is intended to model, e.g., an eavesdropper on the local area network of the user, such as an administrator monitoring web usage at a local firewall. However, if the same LAN also serves the end server, then the eavesdropper is effectively global, and we provide no protections against it.

Shlomi Dolev and Rafail Ostrobsky: Xor-trees for efficient anonymous multicast and reception [39]

There are restricted adversary and more realistic adversary defined. Their short characteristics would read as:

1. Restricted adversary: (outside adversary) who is allowed to monitor communication channels but is not allowed to infiltrate/monitor internal states of any processor in the network.

2. More realistic adversary: (internal) adversary that can monitor all the communication between stations and which in addition is also trying to infiltrate the internal nodes of the network. Adversary ... manages to compromise one or more of the internal nodes and is therefore capable of examining every message and all the data that passes through this infiltrated node. Internal k-listening adversary – infiltrates k nodes.

Oliver Berthold, Hannes Federrath, and Stefan Kpsell: Web MIXes: A system for anonymous and unobservable Internet access [7]

This paper primarily deals with traffic analysis. Passive attacker here

- can eavesdrop on communication links but cannot modify any network traffic,

- controls up to $n-1$ MIXes from a cascade,
- controls cache-proxy and knows content of all messages and the receiver (cache proxy serves unencrypted data),
- block any message, generate own messages, and modify messages.

Ian Goldberg: A Pseudonymous Communications Infrastructure for the Internet [45]

The auther defines a set of threats instead of attackers. The list contains the following items.

1. Web site operator – use of cookies to track the user.
2. Systems Administrators and Internet Service Providers – reads users mail, watches network connections (such as web browsing), and generally monitors all online activities. A sysadmin can read any files stored on network drives, and may also be able to access all the files on desktop or laptop computers.
3. Search engines – discover information about people that they may have placed online.
4. System crackers – use search engines, trojan horse software, and network monitoring (much like a sysadmin) to gather information about someone. They may break into non-public databases.
5. National inteligence – wide net *vacuum cleaner* operations to gather huge amounts of electronic information based on keywords and header information. (e.g. Echelon system). They may engage gathering information from colleagues and acquaintances of people, or in technical attacks using e.g. hidden microphones to gather information.
6. Litigious Groups – threatening and filing lawsuits thus forcing AIP operators, for example, to reveal any stored or logged information they may possess.
7. Organized Crime – attempt to either subvert the network, or the privacy of a nym. Likely to use physical violence for employee subversion, theft, or breaking and entering. May be better funded and equipped than police forces.

Roger Dingledine and Paul Syverson: Reliable MIX Cascade Networks through Reputation [36]

This paper again lists possible attacks on reputation system to better parameterise the attacker's strength.

1. Have enough nodes to own an entire cascade – it allows her to follow messages all the route to the recipient.
2. Gain high reputation to read more traffic – as the system here utilizes reputation, the attacker may try to attract more traffic by increasing her reputation.
3. Replay attack, message delaying – local active attack based on *time manipulation* of messages.
4. Trickle attack –
5. Intersection attack – using MIX cascades rather than free routes helps to defend against intersection attacks from very large adversaries.
6. Influence cascade configuration externally – active external attack. The influence may be realised through disinformation, changes in configuration files, ...
7. Compromise the cascade configuration server – local active attacker targetting most important component of the system.
8. Knock down uncompromised cascades to get more traffic – active, probably global, attack.

Dingledine, Sassaman: Attacks on Anonymity Systems: the theory [34]

This is a lecture given by authors at Black Hat conference in 2003. There is a lot information about attackers and possible threats. The general categorisation is as follows:

- External (wires) or Internal (participants)
- Passive or Active
- Local or Global
- Static or Adaptive

Examples of attackers may be global passive adversary (watches all links), rogue operator (runs one or a handful of nodes), or external attacker (can inject/modify some traffic).

They specially talked about Mixminion system that should be able to protect against all three types of attackers. These attackers are able to launch the following attacks:

- Fooding attack – you know all but one message in the batch.
- Pooling attack – not all messages come out at each flush. Keep a minimum number in the pool, always. Now its harder to target an individual message.
- Trickle attack – what if youre the only one who sends a message into the node in a given interval? More broadly, what if youre the only one who sends a message into the whole network, in that interval?
- Passive subpoena attack – Eve can record messages for later subpoena She can also recognize her own messages, which helps with flooding attacks Fix: Link encryption with ephemeral keys (rekeyed every message / few minutes).
- Active subpoena attack – Mallory can still record messages from the node she runs, and arrive later with a subpoena. Fix: Periodic key rotation.
- Partition attack on client knowledge – adversary can distinguish between clients that use static node lists and clients that frequently update from the directory servers. Fix: Clients must all use the same algorithm for updating from the directory servers. Directory servers must be part of the spec! Directory servers can be out of sync; evil directory servers can give out rigged subsets to trace clients. Fix: DSs must successively sign directory bundles; a threshold of servers is assumed good.
- Partition attack on message expiration date – delaying a message a few days will push its exp date to one end of the valid window – so they wont be uniformly distributed. Fix: No expiration dates. Keep all hashes until key rotates.
- Tagging attack on headers – Mixmaster headers have a hash to integrity-check the fields for that hop. But it doesnt check the rest of the header.

So we can flip some bits later in the header, and if we own the node later in the path that corresponds to the header we just broke, we can recognize the message. We must make the hash cover the entire header.

- Tagging attack on payload – flip some bits in the payload, and try to recognize altered messages when theyre delivered. Fix: Make the hash cover the payload too.

- Multiple-message tagging attacks – If Alice sends multiple messages along the same path, Mallory can tag some, recognize the pattern at the crossover point, and follow the rest. Only works if Mallory owns the crossover point. Fix: Alice picks k crossover points (and hopes Mallory doesnt own most of them).

These are all the attacks that Mixminion can cope with. The authors also mentioned two serious attacks they do not know any defence against.

- Open problem: trickle attack on directory servers Malicious nodes can hold a message and release it later, when circumstances are different. More broadly, were still in an arms race against flooding and trickle attacks.

- Open problem: long-term Intersection attack The fact that not all users are sending messages all the time leaks information. By observing these patterns over time, we can learn more and more confidently who is sending mail, to whom, when, etc. Major unsolved problem in anonymity systems.

They also particularly mentioned conomics of anonymity. The important difference from any other security property is that anonymity is not a question for just one user. Unlike e.g. encryption, its not enough for just one person to want anonymity: the infrastructure must participate as well.

Anonymity systems need cover traffic (many low-sensitivity users) to attract the high-sensitivity users. Most users do not value anonymity much and it results in weak security (fast system) can mean more users which can mean stronger anonymity. High-sensitivity agents have incentive to run nodes so they can be certain first node in their path is good to attract cover traffic for their messages.

Unlike anywhere else, there can be an optimal level of free-riding that improves security properties of anonymity systems.

Michael J. Freedman and Robert Morris Tarzan: A Peer-to-Peer Anonymizing Network Layer [42]

Anonymity against malicious nodes: Tarzan should provide sender or recipient anonymity against colluding nodes. That is, a particular host should not be uniquely linkable as the sender (recipient) of any message, or that a message should not be linkable to any sender (recipient). Authors consider these properties in terms of an anonymity set: the set of possible senders of a message. The larger this set, the more anonymous an initiator remains.

These properties imply the weaker relationship anonymity: an adversary should not be able to identify a pair of hosts as communicating with each other, irrespective of which host is running Tarzan.

Anonymity against a global eavesdropper: An adversary observing the entire network should be unable to determine which Tarzan relay initiates a particular message. Therefore, a nodes traffic patterns should be statistically independent of it originating data traffic.

Sharad Goel, Mark Robson, Milo Polte, and Emin Gun Sirer: Herbivore: A Scalable and Efficient Protocol for Anonymous Communication [44]

This papers gives a list of possible attacks the authors consider important for the design of their system.

- Collusion and Occupancy Attacks – If malicious nodes gather in a clique, they may share information to compromise the anonymity of an honest node.
- Sybil Attacks – A small number of malicious nodes may attempt to overwhelm the system by joining a large number of cliques under many false identities.
- Topology Attacks – We rely on a variant of the Chord distributed hash function to provide the mapping from clique keys to clique members. We treat the security of Chord as a separable problem and do not analyze its resilience against attacks.
- Intersection Attack – By identifying long-running transactions and monitoring clique membership, an attacker may try to subvert a nodes anonymity with an intersection attack.
- Statistical Analysis – Very long-lived transactions in Herbivore are susceptible to statistical analysis: If an attacker observes that node

n_v is disproportionately often in a clique that contacts a certain network service (e.g. a specific web site), then the attacker has statistical reason to believe that n_v is in fact the node contacting that service. While every packet transmitted in Herbivore is anonymized among members of a clique, Herbivore does not protect transactions that are significantly longer than clique lifetimes.

- Coordinator Attack – Since we use a star topology for intraclique transmissions, the central node represents a point of vulnerability. If the central node is compromised ...
- Exit Attack – A malicious node could attempt to nullify the votes cast during the exit phase to incite participating nodes to leave the clique during a long-lived transaction.

Beimel, Dolev: Buses for Anonymous Message Delivery [5]

This paper takes a bit more cryptologic point of view and it defines two types of adversaries for the sake of a security analysis.

- Listening adversary: monitors all communication links and also monitors internal content of some of the processors
- Byzantine adversary: this one is like listening adversary + is able to control some of the processors. It is non-adaptive

Marc Rennhard: MorphMix – A Peer-to-Peer-based System for Anonymous Internet Access (Dissertation) [67]

This one is a Ph.D. thesis. However, the attacker model is not really much elaborated. The author defines two types of attackers: global passive external attacker and partial active internal attacker. He defines realistic threat model for both types of attackers.

He further states that the global observer may be a threat in mix networks with no more than a few mixes. But if the number of mixes grows and the mixes are spread over the world, it is very unlikely any adversary can observe more than a small subset of all mixes. It is of course difficult to prove that global eavesdroppers are no threat to a large mix network if its mixes are spread over the whole planet, but "he has given strong arguments to support this".

Andrei Serjantov and Steven J. Murdoch: Message Splitting Against the Partial Adversary [74]

Similar taxonomy of attackers with one interesting type.

- Global Passive Adversary
- Adaptive Adversary – A non-global adversary can achieve the same results as a global adversary by being able to move points of monitoring, taps, fast enough. This is more powerful than only monitoring inputs and outputs to the mix network. In the global passive and adaptive adversary scenarios, intersection attacks are extremely difficult to defend against.
- Partial Adversary – By relaxing the threat model, we can show what users can do to avoid their anonymity being compromised by a realistic adversary. We consider one particular type of partial adversary. The partial adversary does not monitor all links. He has a limited number of taps and may put these at some, but not all, points on the network.

We intend to improve Alices anonymity against an adversary who can monitor a single anonymity server (AS) (for example, a curious ISP or a corrupt law enforcement officer abusing his subpoena powers). We assume that the ability to observe multiple ASes is significantly more difficult than observing a single AS, because most ISPs do not control multiple ASes, and because law enforcement will be less willing to face the increased accountability and risk associated with obtaining multiple unapproved subpoenas.

We divide attacks into intra-network attacks and endpoint attacks. Endpoint attacks on low-latency networks are the most straightforward: an adversary observing both Alice and Bob can quickly learn that they are communicating. By requiring the path from Alice to the anonymity network and the path from the anonymity network to Bob to traverse separate ASes, we can prevent all of these observed transactions as long as the ASes do not collude.

Intra-network attacks on low-latency networks can also be useful. In particular, paths in Tor and the (no longer deployed) Freedom protocol are generally 3 hopsshort enough to maintain usability, but not so short that two nodes can be certain of linking Alice to Bob if they decide to collude. An adversary who can observe two links on the path breaks this assumption. If such an adversary is common, these designs should reconsider path length.

A successful endpoint attack against a high-latency system like Mixmaster takes a lot more time and effort than one against a low-latency system like Tor. However, because an observer of even a few Mixmaster nodes may be able to link Alice to her recipients over time, our work is also relevant for

protecting such high-latency systems from a single-AS adversary. Further, intra-network observations (particularly during periods of low traffic) can be combined with active attacks such as message flooding to shrink the set of messages that mix with Alices message.

George Danezis and Jolyon Clulow: Compulsion Resistant Anonymous Communications [27]

This paper analysis one particular situation – compulsion. Parties under compulsion could be asked to perform some particular task, which bears some similarity with subverted nodes. For example, this is an issue for electronic election protocols where participants might be coerced into voting in a particular way. Note that compulsion and coercion cannot be appropriately modeled using the concept of subverted nodes from the traditional threat model. The party under compulsion is fundamentally honest but forced to perform certain operations that have an effect which the adversary can observe either directly or by requesting the information from the node under compulsion

Claudia Daz: Anonymity and Privacy in Electronic Services [31]

The author here analyses effects of two basic types of attackers: global passive adversary and local active adversary performing blending attacks on different types of mixes.

Chapter 5

Privacy primitives

If we see privacy in the digital environment as a mean to prevent unintended leakage of information, this information has to be protected by technical means. These technical means are built from privacy primitives that are based on cryptographic primitives or equivalent to these. We can differentiate privacy primitives according to the following criteria:

1. **The parties involved** Who is involved and what are their functionalities/abilities?
2. **The purpose** What privacy goal(s) does the primitive achieve for what information?
3. **The attacker model** Against whom should the information be kept private?
4. **The security-level** Is information theoretic resp. unconditional or cryptographic resp. computational security reached?

We present a classification of privacy primitives following the criteria above in the following sections. These primitives can be used to build larger privacy systems as they will be presented in the following chapters 7 and 6 for the communication and application level. But already the primitives often make use of each other and become stand-alone privacy systems as it will be outlined in this section.

We do not intend to write lecture notes on cryptography here so the summaries are pretty short just to introduce the schemes that are used in anonymity systems. Any implementation of anonymity system necessarily uses these cryptographic schemes so at least a brief overview is given.

Detailed information of these concepts can be found in any book about cryptography.

5.1 Concelation or encryption schemes

Concelation or encryption schemes protect the confidentiality of the content of a text (but they do not protect communication-conjunctures if this text is sent, for instance who sends it from, where, when, to whom). There are two types of encryption schemes, symmetric and asymmetric one. Both types have three phases (key generation and possibly distribution, encryption, decryption):

One symmetric private key for encryption is created and distributed at least to the encryptor and to a possible decryptor in the first phase of symmetric encryption schemes. In the second phase, he encrypts the content to protect with this key. And in the third phase the decryptor (who might be the same person as the encryptor) decrypts the encrypted content).

In asymmetric encryption schemes in the first phase a pair of public and private key is created by the decryptor who distributes the public key to possible enryptors who want to send messages to him. In the second phase, such an encryptor encrypts the content to protect with this public key. Finally, the decryptor who holds the private key decrypts the encrypted content with it in the third phase.

1. **The parties involved:** There are an encryptor and possible decryptors of the signature.
2. **The purpose:** Thereby both types of encryption schemes reach the following two properties:
   - Confidentiality of the content.
   - Unlinkability of encrypted and decrypted content.
3. **The attacker model:**
   - No attacker can break the confidentiality of the content (as long as the encryption scheme is not broken in the asymmetric case).
   - None except the holder(s) (in the symmetric scheme there might be more than one) of the private key knows the linkability of encrypted and decrypted information.

4. **The security-level:** There exist numerous implementations for encryption schemes; the most popular symmetric one might be the one-time-pad and the most popular asymmetric one RSA [70]. Depending on the implementation secrecy and unlinkability can be information theoretic or computational secure.

5.2 Secret Sharing

Secret sharing was invented independently in [75] and [9] and means protocols for splitting secrets into several parts, called shares, which are distributed amongst several participants. The secret can only be reconstructed when a certain number of the shares are combined together; individual shares do not reveal any information on the secret. Let

- n be the total number of desired shares,
- k be the minimum number of shares from which reconstruction is possible,

These are the parameters for a so-called (k, n) secret sharing scheme.

1. **The parties involved:** There are
   - a so-called dealer, i.e., the initial owner of the secret,
   - shareholders, i.e., the participants who get shares,
   - and a reconstructor, i.e., the party that reconstructs the secret.
2. **The purpose:** For the secret that has to be protected secret sharing reaches a balance between the following two properties:
   - Availability: even if some shares are lost, the secret is not.
   - Confidentiality: an adversary who gains access to only a few shares has no advantage in guessing the secret.
3. **The attacker model:**
   - Regarding availability reconstruction works correctly, if dealer, reconstructor and the k shareholders participating in the reconstruction are honest and the communication between them is authentic.

- Regarding confidentiality any $k-1$ shareholders gain no information about the secret as long as the dealer and the reconstructor are honest and the communication between them is confidential.

4. **The security-level:** Shamirs system [75] based on polynomial interpolation is information theoretic secure.

5.3 Commitments

A commitment scheme allows one party to commit to a secret (fix it so that it cannot be changed) without telling another party about it for a certain time. After telling the other party the secret this party is able to verify that this was the secret the first one committed to. Commitments were first invented as unnamed primitives in other protocols, e.g., zero-knowledge proof systems, and only later recognised as something that deserves a name because it occurs so often. The first systematic treatment can be found in [12].

1. **The parties involved:** There are a so-called committer and a recipient of the committed secret.

2. **The purpose:** Commitment schemes have a first phase after which the committer is committed to a secret, but the recipient cannot see it yet, and a second phase for opening and verifying the commitment. Thereby it reaches the following two properties:

   - committing property: The committer cannot change the secret after the first phase.
   - confidentiality property: The recipient does not learn anything about the secret during the first phase.

3. **The attacker model:**

   - Regarding the committing property even a dishonest committer cannot open one commitment in two different ways.
   - Regarding the confidentiality property the first phase does not give the recipient any information about the secret.

4. **The security-level:** The committing and confidentiality property can each be required with either information theoretic or computational security. But no scheme can fulfil both properties with information theoretic security because after the commitment, there are either

two possibilities to open the commitment, then it is not information theoretically committing, or only one, then it is information theoretical confidential.

5.4 Zero-knowledge proofs

Zero-knowledge proofs were first presented in 1985 by Shafi Goldwasser, et al. [47]. With a zero-knowledge proof one party is able to prove to another party that a statement he made is true, without revealing anything other than the truth of the statement.

1. **The parties involved:** There are a so-called prover of the statement made and a verifier of the proof.
2. **The purpose:** A zero-knowledge proof should fulfil the following properties:
   - Completeness: The prover can convince the verifier of correct statements.
   - Soundness: Not even a dishonest prover can convince an honest verifier of wrong statements.
   - Zero-knowledge: None who interacts with the prover gets any new knowledge about her statement except he explicitly reveals information on it.
3. **The attacker model:**
   - If the prover is dishonest and his statement is false he cannot convince the honest verifier that it is true.
   - Even if the verifier is dishonest he cannot learn anything other than the truth of the statement proved by the zero-knowledge proof.
4. **The security-level:** The properties can hold with information theoretic or computational security.

5.5 Blind signatures

Blind signatures enable parties to sign a secret without getting any knowledge about what they sign. In first place this might sound completely pointless, because someone should be interested in what he signs. But for some

applications like anonymous payment systems a signer is not interested in what he signs he even must not be interested in it.

1. **The parties involved:** There are a signer and a recipient and possible verifiers of the signature.

2. **The purpose:** In contrast to traditional signature schemes blind signature schemes have five instead of three phases: In the first phase (the key generation and distribution) the signer creates public and private key for a digital signature scheme and distributes the public key. In the second phase (the blinding) in contrast to traditional signature schemes the text to be signed, is generated by the recipient of the signature (not by the signer) who blinds it (usually by encryption) and sends it to the signer. In the third phase (the signing) the signer signs the blinded text and sends it to the recipient. In the fourth phase (the un-blinding) only the recipient knows how to un-blind the original text and is able to transform the signature to the blinded text to a signature to the un-blinded text. In the fifth phase (the verification) everyone who knows the signer's public key can verify if the signature fits to the text. Thereby blind signatures reach the following two properties:

   - Unlinkability of blinded and un-blinded text as well as unlinkability of the signatures to them.
   - Integrity of blinded text and un-blinded text by the signatures on them.

3. **The attacker model:**

   - None except the recipient knows the linkability of text and blinded text resp. blinded signature and signature.
   - No attacker can break the integrity of the text resp. blinded text as long as the signature scheme is not broken.

4. **The security-level:** Blind signature schemes can be implemented using a number of common signature schemes e.g., RSA [70] and reaches the same security level regarding integrity, this means computational security. Unlinkability can be reached with informational-theoretic security if the encryption used for blinding is informational-theoretic secure.

5.6 Pseudonymous convertible credentials

A credential system is a system in which users can obtain credentials from organizations and demonstrate possession of these credentials. Credentials usually are assigned to pseudonyms. With convertible credentials the users are able to transform a credential issued to one of her pseudonyms to another one of her pseudonyms. This concept was introduced in [17].

1. **The parties involved:** There are users and organizations.
2. **The purpose:** In an anonymous credential system organizations know the users only by pseudonyms. An organization can issue a credential to a pseudonym, whose holder can convert this credential to another pseudonym of hers. Then she can prove possession of this converted credential to another organization and the following properties hold:
   - Integrity of the converted credential.
   - Unlinkability of credential and converted credential and thereby unlinkability of the pseudonyms they are used with.
3. **The attacker model:**
   - Regarding integrity it should be impossible for a user and other organizations to forge a credential of another organization for the user, even with an adaptive attack on the respective organization.
   - Regarding unlinkability an organization cannot find out if two pseudonyms belong to the same user as far as the user does not tell it.
4. **The security-level:** There exist several possibilities of implementation. In [14] a credential system based on the strong RSA assumption and the decisional Diffie-Hellman assumption (that makes the system computational secure) is presented.

In [14] such a credential system is called anonymous. This term might be misleading because the system does not reach anonymity directly, but only pseudonymity by the use of pseudonyms and unlinkability . This might result in anonymity, but does not necessarily do if person pseudonyms are used.

5.7 Pseudonyms

As already outlined in section 2.2 pseudonyms are an important privacy primitive. They act as identifiers of subjects or sets of subjects. Whereas anonymity on the one hand and unambiguous identifiability on the other are extreme cases with respect to linkability to subjects, pseudonymity comprises the entire field between and including these extremes [62].

1. **The parties involved:** There are the holder of the pseudonym and the parties he uses his pseudonym with.
2. **The purpose:** Important properties of pseudonyms can include [22]:
   - Proof of holdership: Digital pseudonyms could be realised as a public key to test digital signatures where the holder of the pseudonym can prove holdership by forming a digital signature which is created using the corresponding private key.
   - Linkability due to the use of a pseudonym in different contexts
   - Convertability, i.e. transferability of attributes of one pseudonym to another: The user can obtain a convertible credential (cf. section 5.6) from one organisation using one of her pseudonyms, but can demonstrate possession of the credential to another organisation without revealing her first pseudonym.
   - Authorisations can be realised by credentials or attribute certificates bound to digital pseudonyms, but also in case of digital vouchers transferable to other people by blind signatures (cf. section 5.5) as well.
3. **The attacker model:**
   - The users can determine the linkability of their pseudonyms themselves.
   - Attacker model of convertible credentials applies to convertability.
   - No attacker can break the holdership of a pseudonym and the correctness of authorisations as long as the signature scheme used is not broken.
4. **The security-level:** The linkability of pseudonyms is absolute. The other properties depend on the primitives used to realize them, usually they are computationally secure.

5.8 Private information retrieval

1. **The parties involved:** The user who queries the database and the server(s) which hold(s) the database and answer(s) his queries.

2. **The purpose:** Private information retrieval (PIR) allows users to retrieve an item from another party (usually by querying a database) without revealing which item he is interested in (privacy for the item of interest).

3. **The attacker model:** Single servers or even collusion of servers do not learn anything about the item of interest depending on the implementation used.

4. **The security-level:** Security depends on the concrete implementation. The only possible protocol that gives the user information theoretic privacy for her item of interest is that the server sends an entire copy of the database to the user. There are two solutions to come to a more efficient solution: one is to make the server computationally bounded and the other is to assume that there are multiple non-cooperating servers, each having a copy of the database.

An early reference that already deals with this problem is [69], but the problem was first formulated unter the name 'private information retrieval' in [21]. Since then numerous solutions have been presented and theoretical bounds calculated how efficient such a system can become under the assumption of no or specific computational bounds of the servers.

5.9 Steganography

Steganography is the old art and the young science of hiding secret information in larger, harmless looking files, the so-called cover data. The main difference to cryptograhy is: If good cryptography is used, the aggressor notices that he cannot understand the cryptotext and will hence presume that the communication is confidential. But if good steganography is used, the aggressor will think that the stegotext is a plausible message which he completely understood. He does not notice any confidential communication.

The young science of steganography uses computers to embed secret data for example in digitalized pictures, video signals or sound signals.

According to Kerkhoffs' principle, the security of steganographic systems must not depend on the secrecy of the steganographic algorithm but on a key

used to parameterized the embedding. Symmetric keys distributed before exchanging secret messages can be used to control the embedding process itself. To increase security, cryptographic systems can be used to encrypt messages before embedding [85]

1. **The parties involved:** There is a sender who embeds the secret message into the cover data, and a recipient who extracts the message.

2. **The purpose:** The purpose of steganographic systems is to hide not only the secret message, but also even its existence. This is helpful if confidential communication is suspicious, unwanted or even illegal.

3. **The attacker model:** An attacker must not be able to decide with probability better than random guessing whether suspected data contains steganographically embedded messages or not.

4. **The security-level:** Information-theoretical security would be possible if a cover is used that already contains the secret message. However, this approach is not of practical relevance. The difficulty in steganography is that stego data must be plausible, i.e., embedding must not significantly change any feature of the cover data. In practice, a steganographic system is considered to be secure as long as there is no algorithm able to achieve the goal of an attacker described above. Among the currently known steganographic algorithms, Perturbed Quantization (PQ) as suggested by Fridrich at al. [43] can be seen as embedding technique which is hard to detect.

Chapter 6

Application privacy

6.1 Techniques for providing privacy in databases

The need for techniques for preserving privacy in public databases (databases that are publicly available) or statistical databases are obvious as these may contain very sensitive information about individuals. Anyone who has access to these databases and has adequate rights for performing queries on data can learn a lot. Even more dangerous are aggregation queries that can combine lots of data together and infer new information that is not explicitly stored in the database. This can be done for both statistical purposes or a purpose of getting information about a particular user(s). Even if the data is anonymized, it is possible to indirectly identify entities by combining some "innocuously" looking attributes.

Statistical databases allow users to retrieve only overall results about some set of entities in the database. Any attempt to retrieve information about any particular entity must be strictly forbidden. As stated above one, can easily conclude that the most important issue in preserving privacy while allowing data to be used for statistical investigations. But these requirements – privacy for the responders and usefulness of the data – are in mutual conflict. Perfect privacy can be achieved by publishing nothing but this has no utility; perfect utility can be obtained by publishing data exactly as received from responders, but this offers no privacy. Data perturbation should permit data analysts (statisticians) to work with the data while preserving privacy of individuals. This section surveys current techniques for both dealing with privacy and data perturbation in statistical and publicly available databases.

Classical and obvious security features (these are not specific for databa-

ses) of public or statistical databases are authorization, identification, and authentication of users who want access the database. Approaches described later would be of no use if there is none of these security features.

We can differentiate between three basic roles of users – standard (or nonstatistical) user who can create, update, and perform queries; statistician (or researcher) who can retrieve only statistical information and an intruder who wants to get some private information and compromise the database. These roles will be used for description of various events that can occur in statistical databases.

From the intruder's point of view there are two main types of attacks. First if an attacker can learn some concrete information about one entity (database was positively compromised) and second if an attacker can conclude that some information (attribute) surely does not hold for some entity (database was partially compromised).

Inference attack on statistical database is a set of database queries that (if properly combined) can reveal a new information (which is not directly accessible) about an entity or a set of entities. Classical situation is that the attacker knows some information about the entity and using this information he tries to learn something new. Suppose that the attacker forms a query which gives only one record as a result. Using this query the attacker can identify one entity. Suppose that the attacker is not allowed to query some other attributes directly but if the database is not well secured he can observe how many results the modified query produces. If the new attribute (e.g. diagnosis=hiv) together with the remaining part of the query produces one result the attacker inferred a new information – this is the case of positively compromised database. If the number of results is zero then the database is said to be partially compromised. Techniques discussed in the next section are to prevent databases from being attacked using this type of attacks.

6.1.1 Overview of main techniques used in statistical databases

[19] provides a nice overview of existing techniques of perturbation and also privacy related issues. This is the main source for this section.

An important issue that also affects security in databases is a type of the database. We can consider online / offline, static / dynamic, centralized / decentralized, dedicated / nondedicated databases. In an online database there is a direct interaction with user while in offline systems, user does not have any control on his queries processing. Static databases do not change in time while dynamic systems do. Centralized systems consist of

only one database, decentralized system can be spread among many sub-databases. Dedicated system is used only for database, while nondedicated shares environment with other services.

Methods that are used for security and privacy protection can be classified into four general groups [1]: *conceptual*, *query restriction*, *data perturbation*, *output-perturbation*. Two models are based on the conceptual approach – *the conceptual model* and *the lattice model*. The conceptual model allows to identify security-related problem on conceptual and data layer. The lattice model describes statistical database information in a tabular form at different levels of aggregation. The aim of this approach is to allow for better understanding of possible aggregation that may reveal some new or redundant information. Methods that are based on the query restriction approach provide protection through one of the following measures: *restricting the query set size*, *controlling the overlap among successive queries* by keeping an audit trail of all answered queries for each user, or *partitioning the statistical database*.

Query restriction approach

This method allows to retrieve statistical data only if the query size (number of entities involved in the query $|C|$ processing) satisfies the condition $K \leq |C| \leq L-K$, where L is the size of the database (number of entities) and K is a parameter that is set by a database administrator (DBA). This parameter K should satisfy the condition $0 \leq K \leq \frac{L}{2}$. It was shown [30] that by using a tool called *tracker* it is possible to compromise database even if K is close to $\frac{L}{2}$. Notice that K cannot exceed $\frac{L}{2}$ because otherwise no statistics would be released.

Query-Set-Overlap Control

Query overlaping is a situation when different queries have many common entities. [38] noticed that many compromises use query sets that have a large number of common entities. This type of control has several disadvantages – cooperation of more users cannot be avoided; there is a need for up-to-date profile for each user and database usefulness may be jeopardized with these limitations. A mechanism that performs comparison between user queries works in $O(L)$ complexity, where L is the size of the statistical database.

Auditing

This is a query restriction method in which a log of queries is saved, and every query is checked for possible data compromise. The given query is allowed or suppressed according to the check result [20]. One problem of this approach is efficiency – the problem of deciding whether a sequence of queries violates privacy was shown to be computationally hard. [55] showed that given a database d and a set of queries, deciding whether an exact answer to these queries leads to full determination of the value of at least one protected database entry is an NP-hard problem.

Partitioning

The main idea of this method is to group individual entities into mutually exclusive subsets that are called *atomic populations* [1]. These are then available for queries of database users. Authors of the method believe that many ways of compromising databases (like if an attacker has quite enough additional information such as when entities are inserted / updated / deleted from the database) can be avoided since atomic population does not contain any information about particular entities. A drawback of this approach, as it was shown in [72], is that many real databases contain tables with only one entity. If these entities aggregate high volume of information, a data loss may occur.

Data perturbation

Data perturbation techniques are divided into two main categories – *probability distribution* and *fixed-data perturbation*.

The probability distribution approach considers the statistical database as a representative sample of a given population with some given probability distributions. The original database is then replaced by a new sample that has the same probability distributions as the original database. Using this method with dynamic databases is very difficult due to the computational overhead because some transformation needs to be made in the transformed sample with every data modification in the original database. This together with possible high inaccuracy (as mentioned below) makes this method not very widely used in statistical databases.

Fixed-data perturbation approach on the other hand changes the values of the attributes, which are to be used for computing statistics, once and for all. This approach often requires another (transformed) database to be created for the statistical purposes only. Data is perturbed in a way

that some random value is added to the real value. This can suffer from high inaccuracies so instead of adding random value this value is multiplied with the original one. Attributes that have binary representation are perturbed with probabilities (*fixed-data perturbation for categorical attributes*) that the value is true or false. Probability value p that is defined by a database administrator is multiplied by the number of entities that satisfy a particular query but without the binary attribute. Advantage of fixed-data perturbation approach is that transformed data can be updated dynamically along with changes in the original data. [53] discuss privacy of random-data perturbation techniques and showed that under certain circumstances this techniques can provide a very little data privacy. They also pointed out some possible directions for new privacy-preserving data mining techniques like *exploiting multiplicative* and *colored noise*.

There is always a problem with accuracy of results with data perturbation techniques. [57] showed that under certain circumstances, 50% bias can occur. [84] provides a study on both the impact of perturbation techniques for protecting databases and the bias problem.

Output perturbation approach

The main difference between this approach and the previous one is that the bias problem is less severe here. This is because the results are based on the original values (not perturbed values) and only the result is perturbed.

First approach is called *random-sample noise* and it was proposed by [29]. The idea is very simple. Set of entities that satisfy requested query is influenced by probability parameter P, that is set by DBA. An entity in the set will be considered in the result with a probability P. The required statistics are computed based on the sample query set. The statistics computed from the sample query set has to be divided by P in order to provide a corresponding unbiased estimator. This method suffers from the resulting inconsistency.

In *varying-data perturbation* approach a random perturbation is added to the query answer, with increasing variance if the query is repeated [4].

Rounding technique takes the result of the query and rounds it up or down to the nearest multiple of a certain base b. There are three types of rounding techniques – systematic rounding, random rounding and controlled rounding. Systematic and random rounding technique add some offset to values in the database. Controlled rounding technique affects more values in the row in such a way that the sum of the row equals to the sum of nonrounded values. Problem with rounding is that it is possible to determine

the true value by averaging the responses to the same query. In general, rounding techniques are not considered to be effective security-tools but they can help if they are combined with some other approaches.

From the methods that were presented above, the random-sample queries method, the varying-data perturbation method, the fixed-data-perturbation method, and the fixed-data-perturbation for categorical attributes are the most promising security-control methods for online dynamic statistical databases. A very good comparison of all methods mentioned here can be found in [1].

So far, we were talking about how data are perturbed to protect private information. One would expect some more precise definition of privacy in context of statistical databases. [37] deals with this issue and authors propose some definitions of privacy, e.g. from the computational point of view. The main contribution of this paper is estimation of a lower bound of perturbation needed to maintain any reasonable notion of privacy.

6.1.2 Private database queries

Publicly available databases can pose a significant risk for privacy of its users, since a curious database administrator can follow the user's queries and infer what the user wants to find out. Users are often cautious about accessing a database when their intentions are about to be kept secret. It can be shown that in the case of a single database, to completely guarantee the privacy of the user, the whole database should be downloaded and queried. [21] investigates whether more efficient solutions to private retrieval problem can be obtained by replicating the database. The paper describes schemes that enable users to access k replicated copies of a database ($k \geq 2$) and privately retrieve information stored in the database. This means that each individual server (holding a replicated copy) gets no information on the identity of the item retrieved by the user.

6.1.3 Data mining

When we discuss privacy issues, we 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, 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 infer-

ence. [80] provides a comparison between "traditional" retrieval of personal information and data mining approaches and [24] discusses on security and privacy implications of data mining. [13] discusses two views of data mining – the desire for privacy by web users and the need of 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 [3, 10, 40].

Even with encryption and other nominal forms of protection for individual databases, we still need to protect against violations of privacy through linkages across multiple databases. [41] presents an overview of some proposals of matching and record linkage methods and points out some essential ideas of these methods (mainly input data representation). Author pointed out problems with large-scale linked databases with respect to privacy of individuals and the need for new computational and statistical technologies for privacy protection.

6.2 E-commerce

[6] did a largescale online shopping experiment to find people's privacy concerns about their personal details. Their findings suggested that under the right circumstances, people easily forget about these concerns and expose their private data. A clustering of the answers revealed four different groups of users:

- privacy fundamentalist – 30 %,
- marginally concerned – 24 %,
- profiling averse –26 %,
- identity concerned – 20 %.

Other important finding was that people do not always act in line with their stated privacy preferences, giving away information about themselves without any compelling reason to do so. This is even more likely when some benefits are offered in return. Privacy statements have no impact on most users' behavior. Authors also suggest how to help users to act accordingly (e.g. by using P3P – The Platform for Privacy Preferences).

6.2.1 E-Cash

Electronic cash is a digital information and is being used to pay the price of a commodity in various scenarios. Electronic cash has been developed to complement the weaknesses of real currency because of electronic commercial trades and has many requirements and security measures to be met so as to be used like real currency. [52] provides a list of requirements compared to what real currency provides.

- Anonymity
  - Real currency provides a user with privacy.
  - If the digital data of electronic cash has or is connected to the information on a user, the cash cannot provide the user with privacy.
- Divisiveness
  - Real currency can be divided or its changes can be offered because the currency has a basic unit.
  - As for electronic currency, the issued data shall be divided.
- Transference
  - Real currency can be transferred to a third party through offering the appropriate amount of money.
  - Electronic currency can also be transferred to a third party through transmitting data; but the security should be kept as at the time of issuing the currency.
- Prevention of double use
  - Real currency cannot be used for the second time unless it is faked.
  - Electronic currency can be used for the second time if the saved information can be copied.

Authors of [52] have analyzed the anonymity of electronic cash in order to come up with effective and safe ways to offer anonymity for the micropayment system. In their new system, anonymity is provided by generating a random number (instead of using blind signatures). The anonymity of the user will be removed in the case of double spending.

[65] presents a privacy protecting e-cash system that provides offline revocable anonymity. This helps in both privacy protection and misuse by criminals. Both bank and the merchant cannot obtain the identities of users but under certain circumstances (suspect criminal activities), the trusted third party (cooperatively with the bank) can remove the anonymity of the transaction and disclose the user's identity. The proposed protocol needs less communication and allows for both coin tracking and owner tracking.

[61] deals with security requirements for off-line E-cash system based on an IC (integrated circuit) card. Off-line E-cash system based on the IC card suffers from overspending, double spending, forgery, altering / eavesdropping transaction content, etc. A lot of technical discussions of the security requirements for theoretical off-line E-cash protocols were done but only a little attention was paid to the security requirements for practical off-line E-cash system (including authentication, key management, implementation of cryptographic algorithms, etc).

6.3 Privacy in location-aware systems

The ability to determine geographical position is a technology with both significant benefits and important privacy implications for users of mobile devices (such as mobile phones and PDAs). Location data can be gathered either internally by the device itself or externally by systems and networks through which devices interact. Geographical data can be stored, used or disclosed under various conditions.

[58] lists out specific privacy issues related to location privacy. These are grouped into eight logical categories. The issues are: collection issues, retention issues, usage issues, disclosure issues, governmental regulation, standard-based regulation, advocacy/public interest group regulation, marketplace regulation. Every issue includes several main interests. For example, the collection issue deals with questions like: " Should users of location-enabled devices be informed when location tracking is in use?", "Should a user be permitted to turn it off?". Privacy implications of location-aware technology must be carefully considered and must provide safeguards necessary to both protect rights of individuals and facilitate evolution of privacy-enabled products and services.

[25] performed an experiment to find out the compensation that people would want in exchange for the information about their actual geographical position. They asked about 1200 participants to price their private information (information about their current geographical position). Using a cover

story, participants were offered to participate in an experiment where they would be tracked for a period of one month for both academic and commercial purposes. They were asked various questions to support the cover story and to fulfill the real purpose of the experiment – how much money people want for their private information. The average price required by the participants (for academic purposes) was 43 EUR and there were about ten percents of participants bidding below 1 EUR. Other details of the experiment as well as the results are in [25].

6.4 Identity management

Since it is now very easy to collect and process huge amounts of personal data for building "profiles" of individuals, it is necessary to allow these individuals to be able to somehow influence what personal data will be processed. PRIME (Privacy and Identity Management for Europe) [48] project has implemented a technical framework for processing personal data. PRIME's vision is to give individuals sovereignty over their personal data so that [15]:

- individuals can limit the information collected about them by using pseudo-identities, certifications and cryptography when performing online transactions.
- individuals can negotiate legally-binding "privacy policies" with their service providers which govern how disclosed personal data can be used and which precautions must be taken to safeguard it.
- individuals and service providers can use automated mechanisms to manage their personal data and their obligations towards data which they have collected from other parties.

To accomplish this, the PRIME project has designed and implemented a practical system-level solution. [15] describes the architecture of this solution.

Rapid growth of online services increases number of different identities a user needs to manage (e.g. a person may have one identity as a bank customer and another identity as a company employee). This leads to a problem with proper identity management and people are not able to control and protect their digital identities against identity theft. [51] discusses the usability and privacy in online identity management solution and proposes a general approach enabling users better control and management of

their digital identities, as well as designs of more secure identity management solutions. [51] also provides a nice overview of existing identity management approaches each illustrated from the perspective of usability and scalability. Their user-centric model solves the scalability problem and has the potential of providing a universal solution while still being compatible with other models described. Their model also provides stronger security than traditional solutions (this is achieved by having a single separate hardware device called PAD – Personal Authentication Device). PAD can take an active part in security transactions.

Another problem is that people have very limited understanding of security and privacy policies when being applied to their confidential information and little control over the manipulation with this information once it has been disclosed to third parties. People perceive and address the related security and privacy issues in different ways, ranging from completely ignoring them to being so concerned to prevent them from using any Internet and web-based applications. Identity management systems and solutions need to simplify users' experience so people can feel they have control over their confidential data and that their data is managed in an accountable way [60].

[60] proposes mechanisms to enforce user privacy by putting users in control and making organizations more accountable. In their model, people use graphical tools to: locally author their disclosure policies in a fine-grained way; obfuscate their confidential data by directly using these policies; associate these policies to the obfuscated data (by creating digital packages). Digital packages can be provided by users to requesters. A requester has to demonstrate to the Tracing Authority that he understands involved terms and conditions. Tracing Authority checks both the integrity and trustworthiness of the requester's credentials. If this satisfies the associated policies, obfuscated data will be disclosed.

Chapter 7

Communication privacy

This chapter concentrates on techniques providing privacy on the network layer with a focus on anonymous communication. Three types of anonymity can be distinguished for communication systems [62]: Sender / recipient anonymity as properties that a particular message is not linkable to any sender / recipient and that to a particular sender / recipient, no message is linkable. Relationship anonymity as the property that it is unlinkable who communicates with whom.

Communication on the network layer is by default not anonymous. The amount of transmitted information identifying sender is quite large and not limited only to IP or MAC address. So if no additional software is used, it is easy for adversary to observe which participants of a network communicate with whom. This chapter will present an overview of techniques that can allow a proper protection of the informational privacy in a network. Additionally to the anonymity techniques presented here, encryption schemes as presented in section 5.1 have to be used to protect not only the communication circumstances but also content of messages.

We can differentiate techniques for communication privacy by the following criteria:

1. **Protection goal** What type of anonymity can be reached (sender, recipient or relationship anonymity)?
2. **Attacker model** Which attackers does the technique (not) protect against (outsiders, participants, providers)?
3. **Trust Model** Whom has the user to trust (providers, participants)?

We give a classification of privacy primitives following the criteria above in the following text.

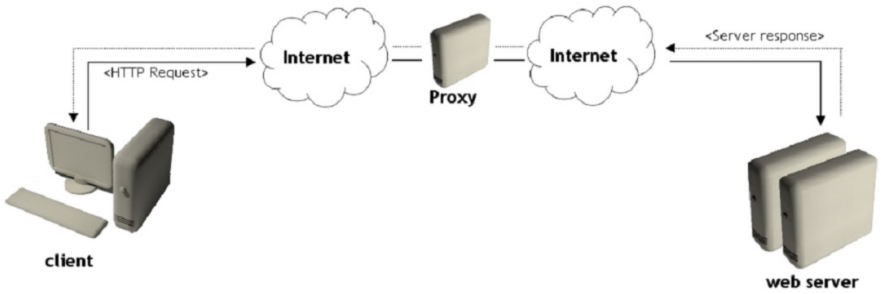


Figure 7.1: Using a proxy for surfing on the Internet

7.1 Simple proxies

One of the most popular concepts for anonymous surfing on the Internet is use of proxies. The main idea behind this technology is that data requests are not sent directly from a client PC to a web server but the client is connected to another server, the so called proxy server, instead. The proxy server retransmits any HTTP-request from the client (shown in Figure 7.1). The web server thus only gets the IP-address of the proxy and not of the client. In addition to that, some proxies also filter out information from the HTTP-request, that could be used to identify the user. These are information like cookies, the operating system used or the browser used. "Active content" like Java script can be blocked, too. In the moment, there are two different ways how to connect to a proxy, either via a website or by using a local proxy. These possibilities can also be combined to form proxy chains.

7.1.1 Website

A website-based proxy allows to use anonymising service without installing any additional software. There is usually a form on such website, where users can fill in the address of the site they finally want to browse. Now the mechanism works as described above: the proxy sends a request to the web server which was specified by the user. After that, the server sends its answer to the proxy and then the requested data is transferred to the client. As an additional feature, the proxy also scans the HTTP-content searching for any link. If there are links, they are transformed in a way, so that they can be used via the proxy immediately. The user still remains anonymous regarding the provider of the websites when using the links.

7.1.2 Local proxies

The second method uses a local proxy. For this approach a software has to be installed on the client's PC, that realizes a local proxy and the proxy has to be registered to the browser as the one to be used when the user tries to connect to a website.

7.1.3 Proxy Chain

There also might be the possibility to use not one proxy, but a chain of several proxies that can be local or external. The combination of several proxies can be useful because different proxies have different pros and cons. By combining several simple but diverse proxies, a powerful proxy chain can be created. On one hand, local proxies can provide a good filter for the HTTP-request and because of being local the speed while surfing in the Internet does not slow down. On the other hand by using different external proxies, the trust necessary in the proxy provider becomes weaker, because the proxy providers have to collaborate in order to recognize to which website the client want to surf.

Protection Goal Sender anonymity against the receiver and relationship anonymity against all others.

Attacker Model What we protect against.

- Protection against the receiver.
- No protection against a single proxy provider or a collusion of proxy providers in a chain.
- No protection against outside attackers who could link ingoing and outgoing messages of one user e.g. by timing analysis.

Trust Model The user has to trust in the proxy, because it can record all transferred information and observe the user's activities. Some proxies insert additional information into the request of the client, e.g. x-forward-for, so that the web server also gets the user IP.

7.2 Crowds

The idea behind Crowds is that the activities of a single user can be hidden within activities of many other users. The system consists of a dynamic collection of users, called a crowd. It was designed to provide sender anonymity while still providing high performance.

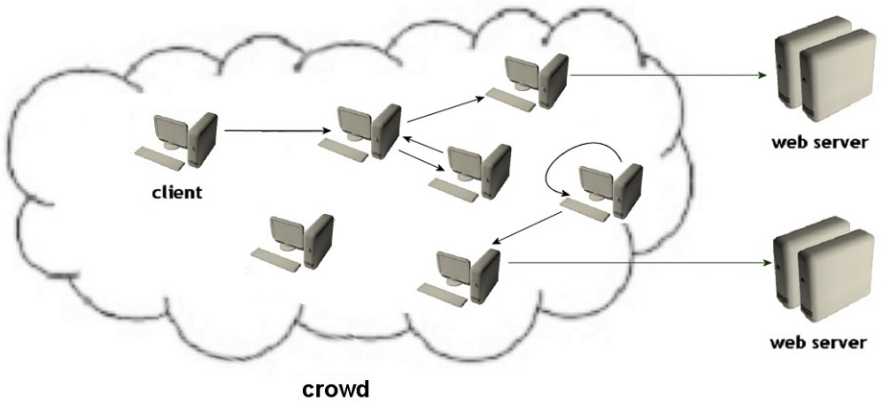


Figure 7.2: Sending request to a web server via Crowds

As discribed in [66] a website request first passes a random number of participants of the crowd before it is sent to the web server. In detail that means, when a member of the crowd gets a message, it decides randomly whether he sends it directly to the destination server or forwards it to another member of the crowd. If the message is sent to another member, he does the same till the message reaches its final destination, the website requested (see Fig. 2). Due to this mechanism neither the server nor a member of the crowd can decide if someone is the initiator of the request or if he only forwards the request. This means plausible deniability is given.

If a user wants to take part in the Crowds-network, he has to install additional software on his local PC, the so called jondo. Furthermore, he has to register with a central server, the so called blender. As one part of the registration procedure, the user has to create a personal login and a password to authenticate himself to the server. The jondo software is a kind of local proxy, so it has to be configured in the browser before it can be used for surfing.

When starting the system, the jondo first contacts the blender to request access to a Crowds network. For access the blender sends him a list of Crowds members (his crowd) and keys for symmetric encryption. The blender further informs all other jondos in the network about the new member. The above mentioned keys are necessary because all information (requests and responds) are encrypted from member to member on the Crowds network. Any web request, coming from the browser is now forwarded to the local jondo, which sends the request randomly to a member of the crowd (this can be another member or even the sender itself).

When a message is sent to or received from a member a special path ID is stored. This ID makes it possible to forward a response of a server back to the requested client. Every jondo saves pairs of path IDs (incoming and outgoing) in a local table. If a jondo receives a message from another jondo, it is checked if the received path ID is already stored in the table. If it is he forwards the message to the next jondo, depending on the second ID of the pair in the table. If the path ID is not in the table, a new destination jondo (or the server) is selected and the message is sent to it. Furthermore a new pair of path ID's is stored in the table: The existing path ID wherefrom the message was received and the new path ID whereto the message is sent.

Protection Goal Sender anonymity against the receiver and relationship anonymity against all others.

Attacker Model No protection against an outside attacker who monitors the whole network (because the jondos just forward messages but do not transform them) or a collusion of Crowds members on the path used.

Trust Model This may be in a form:

- a central instance called blender is used in this system and must be trusted.
- the jondo who receives a message has to forward it to guarantee availability of the network.
- the other members of one's crowd should not collaborate.

7.3 Broadcast

Broadcast is a simple technique that already exists to distribute information in communication networks e.g. for radio and television reception. All participants of the network receive all information sent and select locally which information are relevant for them e.g. by choosing a single television or radio channel. This makes it impossible for observing attackers to gain information about the receiver of special information.

If a specific participant of a distribution network should be addressed by a message implicit addressing can be used. This means there is no link between this implicit address and the physical recipient for anyone than the recipient himself. The address can only be interpreted by the receiver and so the sender did not know concrete information about the receiver. An implicit

address e.g., a large random number has to be sent with the corresponding message and every station receiving the message compares it with his own implicit addresses if he is the intended recipient of this message. If the address is visible for everyone it is called open implicit addressing. To avoid that different messages of the same recipient can be linked by others open addresses should be changed for every message. The message itself shall be encrypted to prevent the other receivers of the broadcast from reading it. If also the address is encrypted this is called covered implicit addressing. But this encyryption forces every station to decrypt all messages to check if he is the recipient.

In a switching network, where every station only receives what the participant requested or another participant sent to him a multicast can be realized. This kind of partial broadcasting means that not every participant in a network receives a message but only a group of them. This reduces the costs of bandwidth needed but the anonymity set decreases as well.

It is possible to use the satellite broadcast network for surfing on the Internet [2]. This kind of broadcast can also be used for file sharing. The broadcast approach allows distribution by sending files via satellite back to the sender. In that case all participants would have an easy opportunity to receive that files.

Protection Goal The requirement may consists of

- receiver anonymity by using implicit addresses.
- receiver unobservability for outsiders if dummy traffic is sent.
- unlinkability of different messages for the same recipient by changing implicit addresses or using covered implicit addresses.

Attacker Model With respect to anonymity and unlinkability protection against observing insiders and outsiders. Regarding unobservability protection against outsiders.

Trust Model If dummy traffic and covered implicit addressing is used no trust in any other participant or provider is needed.

7.4 RING-Network

In a RING-Network the stations are circular cabled (see figure 7.3(a)) and only for local networks suitable. If a station sends a message this message is sent in succession to every station in the RING at least once. By using digital

signal regeneration in every participating station a message is regarding the analogue characteristics independent to the original sender. Every station regenerates the message that it looks like this station is the one that initiated it. This method provides anonymity of the sender against attackers who observe or control stations or connections of the RING-Network. By sending the message trough the whole RING the recipient becomes anonymous as well. A further precondition in order to guarantee anonymity of the sender is that the sending permission is granted without time limit.

If two stations of a RING-Network try to observe the station between them without collaborating, they will not observe anything significant because outgoing messages are encrypted and if implicit addresses are used they cannot be interpreted. So an aggressor must encircle a single station and compare the incoming and outgoing signal patterns. If not, it can only be specified that someone of a group of coherent stations had sent or received a message but not the exact station.

In order to ensure that messages are received by the intended stations it suffices that the sender gets the message back unmodified after one circulation. Because of the serial connection of the stations, all connections and stations have to work properly so that communication of two stations is possible. Defective stations have to be removed from the RING-Network. A braided ring is a possible solution to avoid interferences. It means that two RING-Networks, which are interdigitated into each other, as presented in figure 7.3(b). The first ring connect neighboring stations and the second one the odd ones. This not only doubles the transmission capacity but also compensates a malfunction or breakdown of a station or connection. In this case the braided ring is reconfigured so that the anonymity of the participants remains protected.

In conclusion a ring connection topology with digital signal regeneration and a technique for anonymous multi-access provides sender anonymity against an attacker who controls many stations.

Protection Goal Security requirements will be

- sender anonymity.
- receiver anonymity by sending messages through the whole ring once.

Attacker Model The attackers we protect against

- protection against an attacker who controls many stations as long as the stations before and after the sending or receiving user do

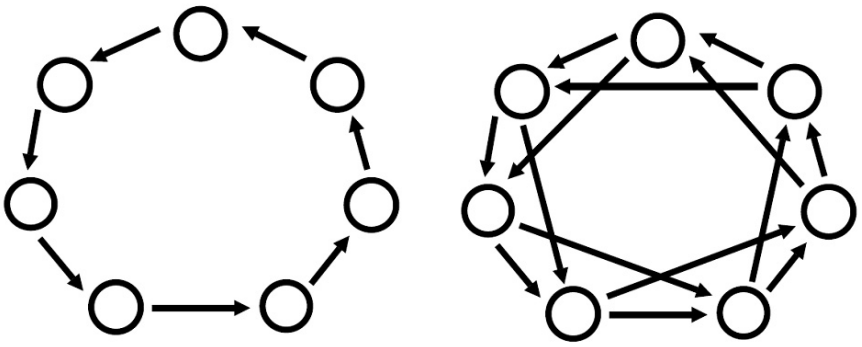


Figure 7.3: (a) ring topology; (b) braided ring

not collaborate.

Trust Model The neighbouring stations of a user must not collaborate.

7.5 Buses

Beimel and Dolev presented in [5] a mechanism for anonymous communication based on so called buses. In their approach every user is modeled as a bus station, while the messages between the users are transferred with buses. The anonymity of the system is based on the idea that a person, who goes by bus in an urban city can hardly be traced by an observer, especially, if the person uses different buses for his journeys.

If a user wants to send a message to another user, he first has to wait until the bus arrives at his station. Then he puts the message in one of the seats on the bus. Beimel and Dolev introduced three types of their system, each with different advantages and disadvantages.

The first type is based on a ring topology and uses only one single bus. As shown in figure 7.4 the bus always moves in one direction. Furthermore, there is a seat for every pair of sender and receiver on the bus. If station A, for example, wants to send a message to station B, it encrypts the message with the public key of B and puts it onto the seat AB of the bus. To ensure that an attacker cannot decide whether a station wants to send a message or not, every station always has to send messages to all other stations, if the station currently has the bus. The attacker cannot decide if there is any "real" communication between the stations at all. To receive messages a station has to decrypt and check all messages in 'his' seats because the others could have put a message in.

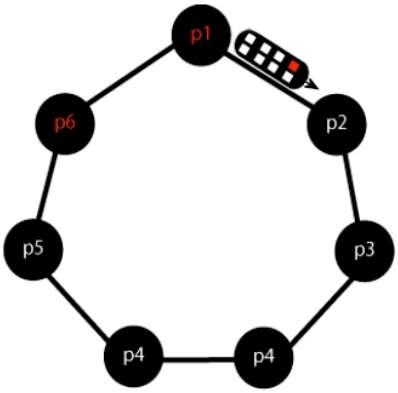


Figure 7.4: Ring network with only one bus present

This has an optimal communication complexity, because only one bus is necessary. But messages need a lot of time to be transferred from its sender to its recipient.

A modification of the system uses variable seats instead of fixed seats. In this case, the sender encrypts his message in an onion-like manner with all public keys of the stations, which the bus will pass on the way to the receiver. The message is encrypted first with the public key of the receiver and after that with the public keys of the stations between the sender and receiver in vice versa direction. Now every station decrypts the incoming message and checks, if the content is meaningful or not. If it is meaningful, the station is the receiver of this message and so the message can be deleted or rather be exchanged by dummy traffic. Otherwise the message is forwarded to the next station. Having no confirmed seats increases the probability of collisions. Therefore the number of the provided seats for the bus has to be calculated suitable.

The second type introduced by Beimel and Dolev uses two buses on each connection between two stations. This leads to a good time complexity but a bad communication complexity. In order to provide a system, with both a good time complexity and a small communication complexity, a cluster concept is introduced as the third type. As shown in figure 7.5 the nodes or stations are integrated in clusters with nearly an equal size. Every cluster has its own bus, to transfer the messages.

In conclusion, buses enable to use the technique of the RING-Networks in a higher communication layer for any network topology. On the one

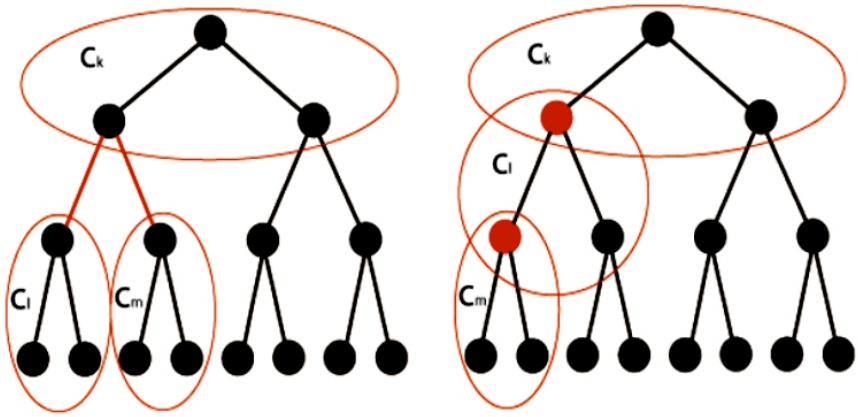


Figure 7.5: Network divided into clusters

hand the approach allows a flexible configuration between communication complexity and time complexity in contrast to ordinary RING-Networks. But on the other hand an implementation as realized in [49] has shown that the system is only usable for relatively small networks and also needs a high amount of dummy traffic to hide meaningful interactions.

Protection Goal Here we have

- sender anonymity.
- receiver anonymity.
- relationship anonymity.

Attacker Model Two types of attackers were described: One the one hand observing attacker who can read messages on the network and control some of the stations. On the other hand manipulating attackers, who can create, manipulate or delete messages.

Trust Model As shown in [5] the system is not secure against DoS attacks. So it must be guarantied that such attacks do not happen.

7.6 DC-Network

This kind of network was specified by David Chaum in 1988 [18]. DC-network can stand for Dining Cryptographers network - an introducing ex-

ample used by Chaum to describe the idea of DC-Networks. It is also possible that DC present the initials of the author David Chaum. The technique is designed for realizing sender anonymity on a variety of communication network topologies.

In order to explain the idea behind the DC-Network the following example is presented: Three cryptographers eat out in their favorite restaurant. After finishing dinner the waiter informs the three cryptographers that the bill has already been paid anonymously. The cryptographers respect this but want to know if one of them paid the bill or the National Security Agency. In order to resolve the uncertainty they use following method: Every cryptographer flips a coin and shows the outcome to the cryptographer on his right. So that every result is only known by two of them and each cryptographer knows two results. Everyone of them compares the two known results and discloses to the others only whether the results are even or uneven. If one of the cryptographers is the payer he would negate his result that means if it is uneven he tells the others that it is even. When the number of the uneven results is also uneven indicates this that a cryptographer had paid. Otherwise none of them is the payer.

By translating this example in a communication network it is called superimposing sending. This technique means that every station sends to a fixed point in time its message or a meaningless one and these messages will be overlaid received by all stations. A station firstly generates secret and random key-characters and communicates one key-character to exactly one other station in the network. These key-characters have to be transferred via a channel that guarantees secrecy. This procedure will be repeated for every station in the network. So, every station has as many key-characters as there are other stations in the network and keeps them secret.

If a station wants to send a message it takes all known key-characters and the message and superimposes them. Therefore a fixed length for the key-characters and the message (use-characters) is required. Superimposing means that all characters are added together byte by byte. This is called the local superimposing. All stations that do not want to send a message have to send an empty message superimposed with all known key-characters.

Every station sends the result of its local superimposition. All the characters that were sent are now being superimposed globally. That means they are subtracted from the local superimposing result. The resulting sum-character is distributed to every station in the network. Because every key-character was added exactly once and subtracted exactly once, and the key-characters hence erase each other after the global superimposition, the sum-character is the sum of all the sent use-characters. If no member

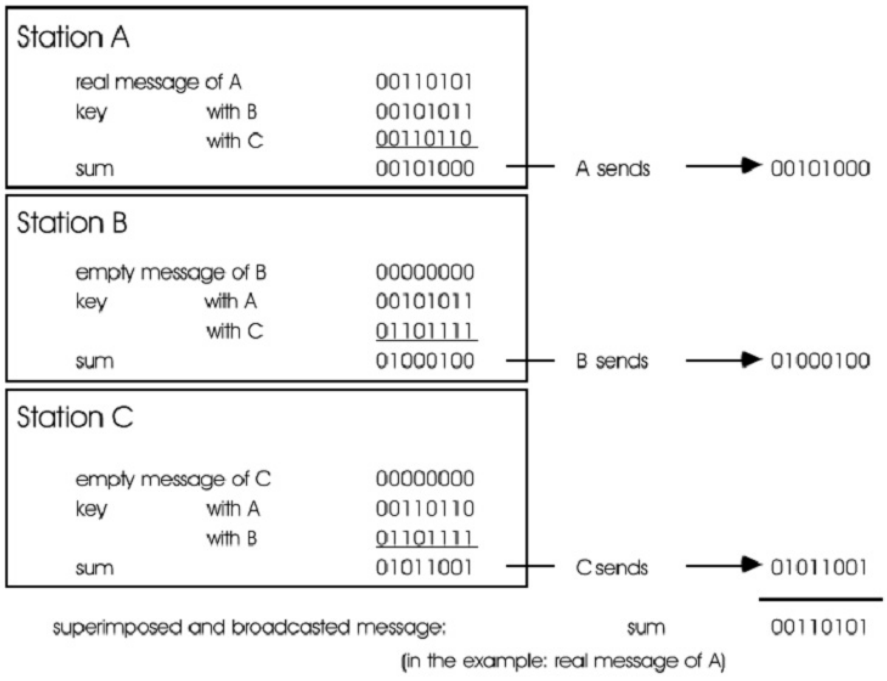


Figure 7.6: Superimposing in a DC-network with three stations

station wanted to send a use-character, the sum-character is the character, which corresponds to 0. But if exactly one member station wanted to send a use-character, the sum-character is equal to the sent use-character.

If the binary characters 0 and 1 are chosen as alphabet, then this obtains the - for practical purposes important - special case of the binary superimposing sending, which was specified by David Chaum. In this case, one does not need to distinguish between addition and subtraction of characters but uses XOR-operation. In figure 7.6 the local and global superimposing is shown for such a binary coded system.

In order to avoid concatenations of messages of one sender the preconcerted key-characters have to be changed for every DC-turn. Otherwise the local sum of a station that sends an empty message would stay identical. The exchange of key-characters can be reduced by using a generator for pseudo-randomly key-characters.

By superimposing sending collisions may occur, if two or more stations of the network want to send simultaneously. All stations will receive the sum of the simultaneously sent characters. But the result will be a meaningless

message. Collisions are a common problem in distribution channels with multi-access. It can be solved by access methods that preserve the anonymity of the sender and also preserve the impossibility to link sending-events.

Every participant of the system gets to know the global sum and consequently the original message. To keep the message content secret as for every anonymising technique an encryption system should be used. Implicit addressing provides receiver anonymity.

The DC-method is very susceptible to denial of service attacks. That means if one station breaks down or has malfunctions only a meaningless message would be transmitted. So the concerted rules have to abide by. Only if everyone transfers its local sum and everyone gets the global sum a DC-Network works fine. But it is a very expensive technique regarding network traffic because with an increasing number of participants also the number of messages and key-characters transferred increases.

Protection Goal There are four of them:

- sender anonymity.
- receiver anonymity by using implicit addresses.
- relationship anonymity.
- sender and receiver unobservability by using dummy traffic.

Attacker Model anonymity and unobservability even against insider attackers, but the system is vulnerable to denial of service attacks.

Trust Model All participants have to abide by the concerted rules.

7.7 Mixes

The idea of Mixes was described by David Chaum in [16]. The method uses public key encryption and was designed for email systems to provide sender anonymity, receiver anonymity and relationship anonymity without the need of a central trusted service. The research into mixes has been quite extensive and so this section may seem to be a bit disproportionate. This is however given by their importance for anonymity systems as we know them today.

In general, Mixes can be understood as a number of proxies one after another. The idea is so far similar to proxy servers like described in section 5.1. In contrast to regular proxies Mixes consider an attacker who can eavesdrop all communications in the network as well as control all Mixes

but one. Mixes have a number of mechanisms, which are described in the following sections.

7.7.1 Basic functionality

As said before, in this approach the clients do not send their requests directly to the web server (or to another destination) but to a so-called Mix. In order to hide, which participants communicate with each other, the Mix does not send the incoming messages instantly to the destination server. Instead of this, the Mix stores several messages form different clients for a defined time, transform the messages (therefore the name Mix) and then forward it to the destination server or to another Mix simultaneously. Due to that, even a global eavesdropper, who can observe all incoming and outgoing messages of the Mix cannot decide which incoming message belongs to which outgoing message. There are a number of Mix building bocks that ensure the security of the system. In almost every approach, which deals with Mixes the basic ideas are used. Only specific implementations can vary from system to system.

7.7.2 Preprocessing: Transforming the message

In order to send a message, the client has to prepare its message. First it has to decide which way the message shall take through the network. That means it has to specify to which Mix the message shall be forwarded before it is sent to the destination server. In order to improve the security of the system it is appropriated to use not only one Mix but several Mixes. In this case it is also important to configure in which order the message shall be forwarded. As a next step the client uses the provided public keys of the Mixes to encrypt its message. Attention has to be paid in this context to the order of the encryption. That depends on the order in which the Mixes will get the message. The whole process can be understood as putting a letter in an envelope, addressing this envelope and then putting it again in an envelope, and so on. So, when the first Mix gets the so prepared message, the Mix will open (or better decrypt) the message and will find an address inside whereto the message has to be send next. This is also shown in figure 7.7.

To ensure that an attacker cannot trace a message from one Mix to another, it is necessary that the messages have no identifying characteristic. One could be their size. One solution is to define a fixed size for all messages. That means short messages have to be filled up with meaningless information

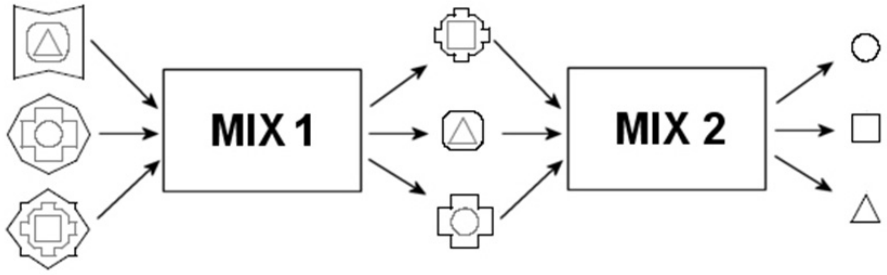


Figure 7.7: Mix cascade with two Mixes

and long messages have to be cut into pieces.

Batch, pool mixing

When a Mix operates in batch-mode, it collects a fixed number n of messages, encrypts and reorders them before all stored messages are forwarded at once. In contrast to that a Mix which operates in pool-mode has always n messages stored in the buffer. If a new message arrives one of the stored messages is picked out and forwarded (see also [54]). Then n is the batch resp. pool size.

Test-for-replay

An often discussed type of attacks is the replay attack (see also chapter XXX). An attacker could copy a message he eavesdropped beforehand and send copies to the mix. These messages would take the same way through the network like the original message, because the decrypting and the sending algorithms work both deterministic. By observing the network, a characteristic pattern of the copied message can be found. These patterns could easily be traced. In order to prevent such an attack, copies of messages have to be identified and filtered out.

One possibility to identify such invalid messages is by using timestamps. The Mix getting a message receives also a tag which tells the mix in which timeframe the message is valid. If the message arrives too late at the Mix, the forwarding will be denied. Another possibility is that the mixes store a copy of every message they already sent. Hence, new messages can be compared to this database. For security reasons it is recommended to restrict the volume of the database to a minimum. Messages should be stored only a short period of time before they are deleted. Another possibility to prevent replay

attacks is the use of a technique called stream cipher. In this case no test for replay is necessary, because a stream cipher must be newly synchronized at any new intrusion. That is because of the fact that the encryption of the present bit depends on the bits processed previously.

Dummy traffic

Even a global eavesdropper should not be able to decide if messages are sent on the network at all, because already the information that specific participants exchange messages can be seen as a lack of security. But simulating for an eavesdropper that not messages were sent on the network is not possible. Instead of this it is possible to send always messages on the network even when no information has to be transferred. This has the same effect as sending no messages, because an eavesdropper cannot decide if a meaningful message is send, or if the message contains only meaningless data. The Sending of such meaningless data on the network is called dummy traffic. According to the idea of Mixes this means for instance that a Mix could randomly forward dummy traffic to another Mix on the network. This mechanism has also a benefit for Mixes, working in batch-mode: Normally these Mixes have to wait until a certain number of messages arrived at the mix, before all stored messages can be forwarded at once. This strategy can tend to create long time of delay, when not enough messages were sent to the Mix. With the help of dummy traffic it is possible to solve this problem. The Mix simply creates dummy message to fill up the buffer.

Return addresses

So far only the principle of for anonymous sending of messages was described. To allow the recipient to reply to such a message anonymous or untraceable return addresses can be used. First of all, the sender X has to send a return address A_X together with his message M to the receiver Y. He also has to send his public key pk_X to encrypt the response and in addition a random key R_1. A mix, who gets such a response, will transform the message like this:

$$pk_X(R_1, A_X), pk_X(R_0, M) \rightarrow A_x, R_1(pk_x(R_0, M))$$

Explanation: [address part], [message part]→ [concrete adress], [encrypted message]

So the Mix decrypts the first part of the response message and finds an address A_X and a key R_1. With the help of the key the Mix encrypts the

message once again and sends it to the indicated address. By using several Mixes instead of only one, the same procedure will be processed at every Mix. In this case the address part $pk_X(R_1, A_X)$ is build up like an onion. That means every Mix, encrypting the message will find a specific R and A. Like in the sending condition, a Mix only knows the next Mix, which it has to forward the message but not the complete way form the sender to the receiver. Finally the original sender receiving the forwarded answer, is able to decrypt the complete message part because he is the owner of the private keys which are can decrypt all R and pk_X.

Checking sender identity

If an attacker blocks the message of a specific participant, this message is isolated from the anonymity group. The same would happen if a message from a specific participant is surrounded by manipulated or generated messages from the attacker. This type of attack is known as $(n-1)$-attack (see also Chapter X). Therefore no general solution exists to prevent this type of attack. One possibility is to provide a mechanism, which makes the Mix able to identify each participant. So a trustworthy Mix can check if the buffered messages were sent by a sufficient number of different users.

Mix-channels

Mix-channels are used to handle a continuous stream of data in real-time or with only small delay through a chain of Mixes. In order to realize a Mix-channel Pfitzmann et al. [63] presented two kinds of channel that are required: the Mix-sending-channel that goes from the sender and ends in a selected Mix (here Mix_m) and the Mix-receiving-channel that starts from the Mix_m and goes to the recipient. The data transfer is coordinated with an asymmetrically encrypted Mix-input-message that contains information about the Mix_m connecting the two channels, and if the user sending the Mix-input-message acts as a sender or a receiver. After that every Mix in the chain can decrypt this Mix-input-message and at the end the plain text is broadcasted to all subscribers. Now the channels can be established using establishment-messages of both participants. They choose the Mixes for the data transfer channel to the Mix_m and keep them private. So everyone only knows half of the way and Mix_m relays the incoming data of the Mix-sending-channel to the Mix-receiving-channel. The two halves of the Mix-channel are necessary to reach anonymity of the two participants against each other. The stream of data is now transferred with link en-

cryption and symmetric encryption through a chain of Mixes. In this chain every Mix assigns the incoming data stream to a Mix-channel and decrypts the stream with its key until it arrives at Mix_m only link encrypted. The Mix-receiving-channel is established but used in the reverse way as the Mix-sending-channel. Thus all data transferred from Mix_m to the recipient has to be encrypted and the recipient has to decrypt the data transferred with all his keys. Every sender/recipient must have the same number of sending/receiving channels otherwise they are observable. So the usage of dummy channels is appropriate. As presented in [63] there are considerations to use this technique for narrow-band ISDN because the slower asymmetrically encrypted establishment of the channel is detached from the data transfer, which is symmetrically encrypted. So each bit of the data transfer channel could be decrypted instantly.

7.7.3 Mix topologies

The concept of Mixes also works with only one single Mix present, but in this case the user completely has to trust this Mix. Therefore not only one but typically a chain of Mixes is used. As already said before, it is sufficient that one Mix of the chain is trustworthy. There are different methods to organize the cooperation within the network. One possibility is that every Mix exists independent in the network and the participants decide freely which way their messages should take. So every node can communicate to all another nodes in the network. This topology is called fully connected Mix network. Another possibility is the definition of a specific chain of Mixes which has to be used. This chain is called a Mix cascade. Besides these two extremes a lot of variations of hybrid systems exist e.g., sparse expander graphs [26].

As mentioned in [8] there is a controversial discussion which of the two Mix-topologies, Mix networks or Mix cascades is the better one.

In the following we give a discussion of some advantages and disadvantages of Mix networks and Mix cascades according to [8] and [26]:

In a Mix networks the user can decide on his own, which Mixes he wants to use for the interaction. This approach provides good scalability and flexibility. Furthermore by selecting the Mixes randomly an attacker does not know which Mixes he has to control in order to observe a message. So the attacker has to control large parts of the network. In contrast to that an attacker of a Mix cascade exactly knows which Mix he has to control in order to observe user messages. Furthermore Mix cascades are vulnerable to denial of service attacks, since disabling one Mix in the cascade will stop the whole system. It is also mentioned in [26] that cascades provide small anonymity

sets in the general case and do not scale well to handle big traffic. Another problem is the high latency of the messages under the condition that the batch sizes of the used Mixes are large. But these arguments only hold if the same types of mixes are considered, especially if mixes run as an application on user's computers. This is certainly true for many mix networks but mix cascades mostly contain mixes on servers that only run this anonymizing services and are able to handle large sets of users. If these mixes are tun by trustworthy organizations these servers will be difficult to be attacked successfully.

On the other hand the authors of [8] found out that Mix networks are vulnerable against powerful attackers, which control all Mixes but one. Also Mix networks are estimated to be weak against blending attacks. As argued by Dingledine et al. [35] this kind of attack has no correlation with the network topology but synchronous batching. Also intersection attacks and traffic analysis have no effect in Mix cascades. Another disadvantage of Mix network is that some Mixes could be used marginally while others are overloaded. In the first case it is necessary to produce a lot of dummy traffic or to wait a long period of time to increase the anonymity set.

Protection Goal Two types of anonymity

- sender anonymity.
- relationship anonymity.

Attacker Model There are two threats:

- protection against powerful attackers who can observe the whole network and control many Mixes (big brother).
- susceptible to denial of service attacks and $(n-1)$-attacks.

Trust Model At least one Mix in a path used in a Mix network or in a Mix cascade has to be trusted.

7.7.4 Existing systems

In this section several existing Mix systems are presented that are or were available for practical use. Thereby we distinguish between low latency and high latency systems.

High latency

Mixminion: Mixminion is based on the specification of the Type-III-Remailer protocol. As described in [28] it enables a user to send and receive e-mails anonymously and thereby also take part anonymously in news groups. The same anonymity set is shared to forward and replay messages. That also means that a remailer cannot distinguish between these two types. A message that shall be transferred is conformed to a fixed size by cutting it into pieces or padding it with dummy data. Mixminion is for asynchronous e-mail conversation so it requires little synchronization and coordination between the nodes. Each packet is send trough a network of Mixminion servers, where users can choose a route.

Mixmaster: Mixmaster was designed for the purpose of anonymous e-mail conversation. Its functionality is based on the type-II-remailer as described in [59]. By sending an e-mail packages of fixed sized are created and each packet can be sent through the mix network via another route. But the last Mix which will send the message to the receiver has to be identical for all packages of an e-mail message. Only this Mix can reassemble the e-mail. A mixmaster server collects messages in a pool and forwards them in random order. If the traffic data is insufficient the mixmaster creates dummy messages automatically. The mixmaster system provides anonymity for sending or receiving e-mails and communication relationships.

Low latency

AN.ON project: AN.ON[1] provides a system that uses the topology of Mix cascades. The user installs on his computer a client software called JAP. After that he can choose between different fixed routes of Mixes for anonymous Internet surfing. All packages that are transferred through a Mix cascade have the same size and are sent in a batch from Mix to Mix. In order to secure from traffic analysis also dummy traffic is used. This provides sender anonymity to users regarding their web surfing.

TOR: TOR [33] is a circuit based anonymous communication service that uses onion routing. It provides support of anonymity for TCP protocol

[1] `http://anon.inf.tu-dresden.de/`

based services like web browsing, instant messaging, e-mail and peer-to-peer. The TOR network consists of several hundred nodes called TOR server. A client chooses a random route of nodes trough the network and builds a circuit. Each node in the network only knows its predecessor and its successor. The data through this circuit can leave the circuit at the end or in midstream so that the observation of the circuit end is unprofitable. The traffic is divided into fixed size cells. Filter mechanism for privacy enhancement are not provided by TOR. Therefore proxies like Privoxy are recommended. The goal of TOR is to maximize anonymity and reduce the latency to an acceptable level.

Freedom Network: The freedom network [11] consists of a private network which is operated by the freedom provider. Freedom supports a lot of Internet protocols for e-mail and http services. In addition it provides services for management of pseudonyms. Packages are sent from the client to the freedom network before they are forwarded to the Internet. The nodes of the freedom network named anonymous Internet proxy and are not linked in a fixed topology. The user can choose the nodes his data shall pass and the order of the nodes. The data stream is scanned, user data like IP address is filtered out and substituted. The response stream is also filtered. That means the local application is not aware of the conducted changes. The freedom network focuses on the speed of data transfer. So no cover traffic is used and the data packages have a variable size.

Tarzan: Tarzan [42] is an anonymous peer-to-peer network based on the IP protocol. By providing a kind of IP tunnel it is independent of a concrete application. It is decentralized and uses an open ended, Internet wide pool of nodes. Every peer in the network can act as a Mix. A message initiator selects a route of peers pseudo-randomly through a restricted topology. At the end of the Mix chain is the network address translator, who changes the origin of the packages and communicates directly with the receiver. So a bridge between the sender and the receiver is created. The sender of a message can be concealed because every participant could have generated a message or only relayed traffic for others. Tarzan makes also use of dummy traffic so they can protect the data against traffic analysis. It protects also against network edges analysis because a relay do not know if it is the first of a mix path. Because of the large number of possible peers the significance of targeted attacks is reduced. Tarzan provides

anonymity against malicious nodes and against a global eavesdropper.

7.8 Private information retrieval

In general it is possible to observe in which content someone is interested because of the IP-address, which users normally left behind while downloading messages from a service provider. In order to avoid that interests can be observed it is theoretically feasible that every user downloads any news on a news server and makes a local news selection. But this may overstrain the news server and it would increase the amount of data that have to be transferred. In order to reduce the costs of bandwidth and also to protect the selection information of a user against an observer the private information retrieval was developed. The idea behind this is as described in [Quelle] some kind of superimposing of information like it has been shown in the section about DC-Networks.

For this technique several servers with exact identical databases are required. In order to conceal which piece of information a user is interested in different requests are made to many servers. These requests are vectors, that are composed of binaries and depending on the digit it notify which information is requested. If a piece of information is requested the vector would contain 1 on this digit if it is not requested it would contain 0. Before sending randomly created requests they are combined with vectors, which represent the piece of information the user is interested in. So after receiving all requested information the vectors are conjunct trough superimposing (XOR-operation) and so only the relevant information is left over. The communication between user und servers is encrypted and the results are decrypted by the user. This technique is similar to the method of the DC-Networks.

But it has to be point out that this is only possible with many servers that receive and store each message in the intended order and adding news must take place simultaneous. So actualising news is complex and difficult. By using this approach an attacker and even a news server would be unable to determine which position in memory is being read and consequently cannot spy the information the user is interested in. If only one of the servers do not cooperate with an attacker it is not possible to determine the interested information of a user.

Protection Goal To hide selection data of a user.

Attacker Model Two types of attacks:

- protection against the provider of news services.
- protection against observing attacks.

Trust Model a collaboration of all news server have to exclude.

7.9 General principles among the systems

Based on the presented systems it can be recognized a number of basic functionalities which are reused in the different approaches. In order to provide communication privacy the following mechanisms are used:

Sender anonymity

- Requests are not sent directly from sender to receiver. They are transferred to other nodes of the system first before they are sent to the receiver.

Receiver anonymity

- A large number or even all participants of the system get the messages which were sent.
- The receiver is not addressed explicitly. Every participant decides locally if the received message is intended for him.

Hiding the communication between sender and receiver

- By sending dummy traffic meaningful communication messages can be concealed.
- The creation of an anonymity set can hide the activities of a single user within the activities of many other users.

In conclusion the anonymity can be improved the more the presented mechanisms and systems are combined. In figure 7.8 an example network is shown which uses a configuration of different approaches. While such a configuration is theoretically possible it would probably be very instable in practice. Even when the configuration works fine there would be a very high latency.

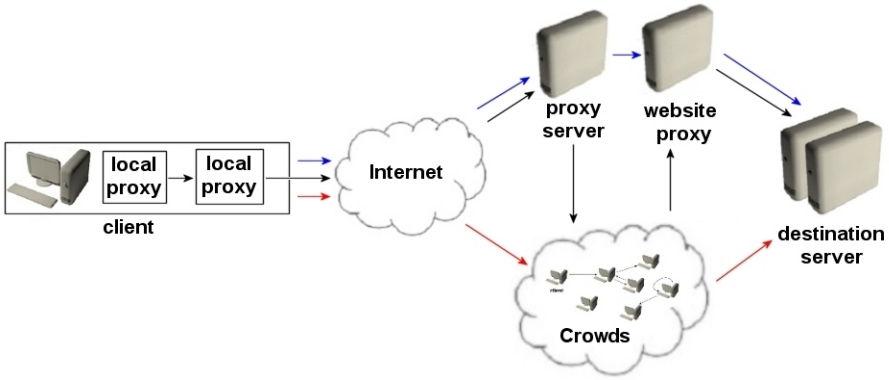


Figure 7.8: example of combining different approaches

Chapter 8

Conclusions

The work presented on preceding pages will provide a solid basis to our follow-up research on privacy models and metrics. It would then be somewhat premature to attempt writing any kind of conclusive remarks before we proceed to the next steps.

That follow-up work, planned for the WorkPlan 4 of FIDIS activites, will aim at providing an overview of existing approaches (most common theoretical tools) for modelling relations of identity related information. That deliverable shall also compare privacy models and tools, providing a critical review of the models, comparisons of their potential, and review of their applicability for measurement or quantitative expression of (the level of) privacy.

Bibliography

[1] Nabil R. Adam and John C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys*, 21(4):515–556, 1989.

[2] André Adelsbach and Ulrich Greveler. Satellite communication without privacy - attacker's paradise. In Hannes Federrath, editor, *Sicherheit 2005*, volume P-62 of *LNI Proceedings*, pages 257–268. GI, 2005.

[3] 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 WEBKDD'99 Workshop San Diego, CA, USA, August 15, 1999*, number 1836 in LNCS. Springer-Verlag, 2000.

[4] Leland L. Beck. A security machanism for statistical database. *ACM Transactions on Database Systems (TODS)*, 5(3):316–338, 1980.

[5] Amos Beimel and Shlomi Dolev. Buses for anonymous message delivery. *Journal of Cryptology*, 16(1):25–39, 2003.

[6] Bettina Berendt, Oliver Günther, and Sarah Spiekermann. Privacy in e-commerce: stated preferences vs. actual behavior. *Communications of the ACM*, 48(4):101–106, 2005.

[7] Oliver Berthold, Hannes Federrath, and Stefan Köpsell. Web MIXes: A system for anonymous and unobservable Internet access. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *LNCS*, pages 115–129. Springer-Verlag, July 2000.

[8] Oliver Berthold, Andreas Pfitzmann, and Ronny Standtke. The disadvantages of free MIX routes and how to overcome them. In Hannes Fed-

errath, editor, *Designing Privacy Enhancing Technologies (PET'00)*, volume 2009 of *LNCS*, pages 30–45. Springer-Verlag, 2001.

[9] G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the National Computer Conference*, volume 48, pages 313—317, New York, NY, USA, June 1979. AFIPS Press.

[10] José Borges and Mark Levene. Data mining of user navigation patterns. In *Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, volume 1836 of *LNCS*, pages 92–111, San-Diego, USA, 2000. Springer-Verlag.

[11] Philippe Boucher, Adam Shostack, and Ian Goldberg. Freedom systems 2.0 architecture. White paper, Zero Knowledge Systems, Inc., December 2000.

[12] Gilles Brassard, David Chaum, and Claude Crepeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[13] Alan J. Broder. Data mining, the internet, and privacy. In *Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, volume 1836 of *LNCS*, pages 56–73. Springer-Verlag, 1999.

[14] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, volume 2045 of *LNCS*, pages 93–118, London, UK, 2001. Springer-Verlag.

[15] Jan Camenisch, Abhi Shelat, Dieter Sommer, Simone Fischer-Hübner, Marit Hansen, Henry Krasemann, Gérard Lacoste, Ronald Leenes, and Jimmy Tseng. Privacy and identity management for everyone. In *Proceedings of the 2005 workshop on Digital identity management*, pages 20–27, Fairfax, VA, USA, 2005. ACM Press.

[16] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2):84–88, February 1981.

[17] David Chaum. Showing credentials without identification. signatures transferred between unconditionally unlinkable pseudonyms. In *Proc.*

of a workshop on the theory and application of cryptographic techniques on Advances in cryptology—EUROCRYPT '85, volume 281 of *LNCS*, pages 241–244, Linz, Austria, 1986. Springer-Verlag New York, Inc.

[18] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[19] S. Chawla, C. Dwork, F. Mcsherry, A. Smith, and H. Wee. Toward privacy in public databases. In Joe Kilian, editor, *Proceedings of the 2nd Theory of Cryptography Conference (TCC'05)*, volume 3378 of *LNCS*, pages 363–385. Springer-Verlag, February 2005.

[20] Francis Y. L. Chin and Gultekin Özsoyoglu. Auditing and inference control in statistical databases. *IEEE Transactions on Software Engineering (TSE)*, 8(6):574–582, 1982.

[21] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *Journal of ACM*, 45(6):965–981, 1998.

[22] Sebastian Clauß and Marit Köhntopp. Identity management and its support of multilateral security. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 37(2):205–219, October 2001.

[23] 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.

[24] Chris Clifton and Don Marks. Security and privacy implications of data mining. In *ACId SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 15–19, Montreal, Canada, May 1996. University of British Columbia Department of Computer Science.

[25] Dan Cvrcek, Marek Kumpost, Vashek Matyas, and George Danezis. A study on the value of location privacy. In *WPES '06: Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 109–118, New York, NY, USA, 2006. ACM Press.

[26] George Danezis. Mix-networks with restricted routes. In Roger Dingledine, editor, *Proceedings of Privacy Enhancing Technologies workshop (PET 2003)*, volume 2760 of *LNCS*. Springer-Verlag, March 2003.

[27] George Danezis and Jolyon Clulow. Compulsion resistant anonymous communications. In *Proceedings of Information Hiding Workshop (IH 2005)*, volume 3727 of *LNCS*, pages 11–25, June 2005.

[28] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a type iii anonymous remailer protocol. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pages 2–15. IEEE Computer Society, May 2003.

[29] Dorothy E. Denning. Secure statistical databases with random sample queries. *ACM Transactions on Database Systems (TODS)*, 5(3):291–315, 1980.

[30] Dorothy E. Denning and Peter J. Denning. The tracker: a threat to statistical database security. *ACM Transactions on Database Systems (TODS)*, 4(1):76–96, March 1979.

[31] Claudia Díaz. *Anonymity and Privacy in Electronic Services*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, December 2005.

[32] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In Roger Dingledine and Paul Syverson, editors, *Designing Privacy Enhancing Technologies (PET'02)*, volume 2482 of *LNCS*, pages 54–68. Springer-Verlag, 2002.

[33] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[34] Roger Dingledine and Len Sassaman. Attacks on anonymity systems: The theory. In *Black Hat conference 2003*, 2003.

[35] Roger Dingledine, Vitaly Shmatikov, and Paul Syverson. Synchronous batching: From cascades to free routes. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2004)*, volume 3424 of *LNCS*, pages 186–206. Springer-Verlag, May 2004.

[36] Roger Dingledine and Paul Syverson. Reliable mix cascade networks through reputation. In Matt Blaze, editor, *Proceedings of Financial Cryptography (FC '02)*, volume 2357 of *LNCS*, pages 253–268. Springer-Verlag, March 2002.

[37] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *PODS '03: Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, New York, NY, USA, 2003. ACM Press.

[38] David Dobkin, Anita K. Jones, and Richard J. Lipton. Secure databases: protection against user influence. *ACM Transactions on Database Systems (TODS)*, 4(1):97–106, 1979.

[39] Shlomi Dolev and Rafail Ostrobsky. Xor-trees for efficient anonymous multicast and reception. *ACM Transactions on Information and System Security*, 3(2):63–84, 2000.

[40] 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.

[41] Stephen E. Fienberg. Privacy and confidentiality in an e-commerce world: Data mining, data warehousing, matching and disclosure limitation. *Statistical Science*, 21(2):143–154, 2006.

[42] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, pages 193–206, Washington, DC, November 2002. ACM Press.

[43] Jessica Fridrich, Miroslav Goljan, and David Soukal. Perturbed quantization steganography with wet paper codes. In *MM&Sec '04: Proceedings of the 2004 workshop on Multimedia and security*, pages 4–15, New York, NY, USA, 2004. ACM Press.

[44] Sharad Goel, Mark Robson, Milo Polte, and Emin Gun Sirer. Herbivore: A scalable and efficient protocol for anonymous communication. Technical Report 2003-1890, Cornell University, Ithaca, NY, February 2003.

[45] Ian Goldberg. *A Pseudonymous Communications Infrastructure for the Internet*. PhD thesis, UC Berkeley, December 2000.

[46] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In R. Anderson, editor, *Proceedings of Information Hiding: First International Workshop*, volume 1174 of *LNCS*, pages 137–150. Springer-Verlag, May 1996.

[47] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *STOC '85: Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 291–304, New York, NY, USA, 1985. ACM Press.

[48] M. Hansen and H. Krasemann. Prime white paper. White paper, Privacy and Identity Management for Europe, July 18 2005.

[49] Andreas Hirt, Michael J. Jacobson, and Carey Williamson. A practical buses protocol for anonymous internet communication. In *Proceedings of the Third Annual Conference on Privacy, Security and Trust*, St. Andrews, New Brunswick, Canada, October 2005.

[50] 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.

[51] Audun Josang, Muhammed Al Zomai, and Suriadi Suriadi. Usability and privacy in identity management architectures. In Ljiljana Brankovic and Chris Steketee, editors, *Fifth Australasian Information Security Workshop (Privacy Enhancing Technologies) (AISW 2007)*, volume 68 of *CRPIT*, pages 143–152, Ballarat, Australia, 2007. ACS.

[52] Seo-Il Kang and Im-Yeong Lee Im-Yeong Lee. A study on the e-cash system with anonymity and divisibility. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *ICCSA, part 2*, volume 3481 of *Lecture Notes in Computer Science*, pages 177–186. Springer, 2005.

[53] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Third IEEE International Conference on Data Mining (ICDM'03)*, pages 99–106, Washington, DC, USA, 2003. IEEE Computer Society.

[54] Dogan Kesdogan, Jan Egner, and Roland Büschkes. Stop-and-go MIXes: Providing probabilistic anonymity in an open system. In *Proceedings of Information Hiding Workshop (IH 1998)*, volume 1525 of *LNCS*, pages 83–98. Springer-Verlag, 1998.

[55] J. M. Kleinberg, C. H. Papadimitriou, and P. Raghavan. Auditing boolean attributes. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 86–91. ACM Press, 2000.

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

[57] Norman S. Matloff. Another look at the use of noise addition for database security. In *IEEE Symposium on Security and Privacy*, pages 173–181. IEEE Computer Society, 1986.

[58] Robert P. Minch. Privacy issues in location-aware mobile devices. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 5*, volume 5, page 50127.2, Washington, DC, USA, 2004. IEEE Computer Society.

[59] Ulf Möller, Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster Protocol — Version 2. Draft, July 2003.

[60] Marco Casassa Mont, Siani Pearson, and Pete Bramhall. Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services. In *DEXA '03: Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, pages 377–382, Washington, DC, USA, 2003. IEEE Computer Society.

[61] Haeryong Park, Kilsoo Chun, and Seungho Ahn. The security requirement for off-line e-cash system based on ic card. In *ICPADS '05: Proceedings of the 11th International Conference on Parallel and Distributed Systems - Workshops (ICPADS'05)*, pages 260–264, Washington, DC, USA, 2005. IEEE Computer Society.

[62] 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.

[63] Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner. ISDN-mixes: Untraceable communication with very small bandwidth overhead. In *Proceedings of the GI/ITG Conference on Communication in Distributed Systems*, pages 451–463, February 1991.

[64] Andreas Pfitzmann and Michael Waidner. Networks without user observability – design options. In *Proceedings of EUROCRYPT 1985*, volume 219 of *LNCS*, pages 245–253. Springer-Verlag, 1986.

[65] Weidong Qiu, Kefei Chen, and Dawu Gu. A new offline privacy protecting e-cash system with revokable anonymity. In *Proceedings of the 5th International Conference on Information Security*, volume 2433 of *LNCS*, pages 177–190, London, UK, 2002. Springer-Verlag.

[66] Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security (TISSEC)*, 1(1):66–92, 1998.

[67] Marc Rennhard. *MorphMix – A Peer-to-Peer-based System for Anonymous Internet Access*. PhD thesis, ETH, 2004.

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

[69] R. Rivest, L. Adelman, and M. Dertouzos. On databanks and privacy homomorphism. *Foundations of secure computation*, pages 168–177, 1978.

[70] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[71] 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.

[72] Jan Schlörer. Information loss in partitioned statistical databases. *Computer Journal*, 26(3):218–223, 1983.

[73] 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.

[74] Andrei Serjantov and Steven J. Murdoch. Message splitting against the partial adversary. In *Proceedings of Privacy Enhancing Technologies workshop (PET 2005)*, volume 3856 of *LNCS*, pages 26–39. Springer-Verlag, May 2005.

[75] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

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

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

[78] 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.

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

[80] Herman T. Tavani. Information privacy, data mining, and the internet. In *Ethics and Information Technology*, Hingham, MA, USA, 1999. Kluwer Academic Publishers.

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

[82] 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.

[83] Michael Waidner and Birgit Pfitzmann. The dining cryptographers in the disco: unconditional sender and recipient untraceability with computationally secure servicability. In *Proceedings of EUROCRYPT 1989*, volume 434 of *LNCS*. Springer-Verlag, 1990.

[84] Rick L. Wilson and Peter A. Rosen. Protecting data through perturbation techniques: The impact on knowledge discovery in databases. *Journal of Database Management*, 14(2):14–26, 2003.

[85] Jan Zöllner, Hannes Federrath, Herbert Klimant, Andreas Pfitzmann, Rudi Piotraschke, Andreas Westfeld, Guntram Wicke, and Gritta Wolf. Modeling the security of steganographic systems. In *Information Hiding*, pages 344–354, 1998.

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

[87] 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.