PRивacy Enabled Capability In Co-Operative Systems and Safety Applications

Deliverable 2

V2X measurement approach
## Control Sheet

### Approval

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepared</td>
<td>All Project Partners</td>
</tr>
<tr>
<td>Reviewed</td>
<td>All Project Partners</td>
</tr>
<tr>
<td>Authorized</td>
<td>Antonio Kung</td>
</tr>
</tbody>
</table>

### Circulation

<table>
<thead>
<tr>
<th>Recipient</th>
<th>Date of submission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Partners</td>
<td>15.04.2009</td>
</tr>
<tr>
<td>European Commission</td>
<td>15.04.2009</td>
</tr>
</tbody>
</table>

### Version History

<table>
<thead>
<tr>
<th>Version number</th>
<th>Date</th>
<th>Main author</th>
<th>Summary of changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.1</td>
<td>08.01.2009</td>
<td>Zhendong Ma (UULM)</td>
<td>Initial version</td>
</tr>
<tr>
<td>v0.2</td>
<td>02.02.2009</td>
<td>Florian Schaub (UULM)</td>
<td>Introduction and Chap. 2</td>
</tr>
<tr>
<td>v0.3</td>
<td>11.02.2009</td>
<td>H. Mikulicz (Oracle)</td>
<td>Industrial standard</td>
</tr>
<tr>
<td>v0.4</td>
<td>19.02.2009</td>
<td>Florian Schaub (UULM)</td>
<td>Structural revision</td>
</tr>
<tr>
<td>v0.5</td>
<td>20.02.2009</td>
<td>Lukas Dölle (UBER)</td>
<td>Data storage perspective 2.4</td>
</tr>
<tr>
<td>v0.6</td>
<td>13.03.2009</td>
<td>Lukas Dölle (UBER)</td>
<td>Measurement approaches in data storage and system</td>
</tr>
<tr>
<td>v0.7</td>
<td>20.03.2009</td>
<td></td>
<td>Re-structure</td>
</tr>
<tr>
<td>v0.8</td>
<td>25.03.2009</td>
<td>Zhendong Ma (UULM)</td>
<td>Comp. Aspects, Metrics for development time and runtime</td>
</tr>
<tr>
<td>v0.9</td>
<td>15.04.2009</td>
<td></td>
<td>Requirement and metrics for data security</td>
</tr>
<tr>
<td>v1.0</td>
<td>26.03.2009</td>
<td></td>
<td>Measurement on data security</td>
</tr>
<tr>
<td></td>
<td>27.03.2009</td>
<td>Guido Tijskens (Oracle)</td>
<td>Security model and PCOs</td>
</tr>
<tr>
<td></td>
<td>07.04.2009</td>
<td>Stefan Dietzel (UULM)</td>
<td>System design and verification approach</td>
</tr>
<tr>
<td></td>
<td>09.04.2009</td>
<td>Antonio Kung (Trialog)</td>
<td>submission version ready</td>
</tr>
<tr>
<td></td>
<td>15.04.2009</td>
<td>Z. Ma, F. Schaub, S. Dietzel (UULM)</td>
<td></td>
</tr>
</tbody>
</table>
# Contents

1. **Introduction**  
   1.1. Motivation .................................................. 7  
   1.2. Relation to other deliverables ............................... 8  
   1.3. Short Review on Measurement Theory .......................... 8  
   1.4. General Approach to Privacy Measurement and Taxonomy ......... 10  
   1.5. Document organization ......................................... 11  

2. **Measurement Requirements and Metrics in IT Systems** ................. 12  
   2.1. Requirements and metrics from Legal and Organizational perspective ... 12  
   2.2. Requirement and metrics from system design perspective ................ 13  
   2.3. Requirements and metrics from Data Security Perspective .............. 15  
       2.3.1. Bell-LaPadula model ........................................... 16  
       2.3.2. Industrial perspective on data security ....................... 19  
       2.3.3. Multi-level security from a business-oriented perspective .......... 20  
   2.4. Requirements and metrics from Communications Perspective .......... 21  
       2.4.1. Privacy in V2X communications ................................. 21  
       2.4.2. Privacy metrics in communication domain ....................... 23  
   2.5. Requirements and metrics from Data Storage Perspective ............. 23  
   2.6. Discussion .................................................... 24  

3. **Measurement Approaches for Co-operative ITS** .......................... 26  
   3.1. Measurement-based Verification Approach .......................... 26  
   3.2. $k$-Anonymity .................................................. 27  
       3.2.1. Introduction .................................................. 27  
       3.2.2. Application to communication domain ........................... 28  
       3.2.3. Application to data storage domain ............................ 31  
       3.2.4. Application to system privacy ................................ 33  
       3.2.5. Interpretation ............................................... 37  
   3.3. Entropy-based Measures .......................................... 37  
       3.3.1. Introduction .................................................. 37  
       3.3.2. Application to communication domain ........................... 39  
       3.3.3. Application to data storage domain ............................ 47  
       3.3.4. Application to system privacy ................................ 50  
       3.3.5. Interpretation ............................................... 50  
   3.4. Metrics on Data Security ......................................... 51  
       3.4.1. User Management .............................................. 51  
       3.4.2. Access Control ............................................... 52  
       3.4.3. Encryption and Masking ...................................... 52
3.4.4. Monitoring .............................................. 53
3.5. Discussion .............................................. 53

4. Measurement through PCOs in Information Flow Analysis 55
   4.1. Methodology ........................................... 55
   4.2. PCOs in Information Flow Analysis .................. 55
   4.3. Measurement Based on PCOs ......................... 57
       4.3.1. \(k\)-anonymity Measurement at PCOs ............ 57
       4.3.2. Entropy Measurement at PCOs .................. 58
   4.4. Measurement of System Privacy ..................... 58

5. Complementary Aspects to Privacy 60
   5.1. Cost of Anonymization ............................... 60
   5.2. Utility Measurements ................................ 61

6. Defining Metrics for System Development Time and Runtime 65

7. Conclusion ............................................... 67

8. Bibliography ............................................. 68

A. Oracle security solutions .............................. 72
# List of Figures

1.1. The relations of D2 to other deliverables .............................. 8  
2.1. Legal and organizational perspective .................................. 13  
2.2. Information flow attack without *-property. [1] ....................... 18  
3.1. Verification approach ....................................................... 27  
3.2. Anonymity value $k$ at each quadrant ................................. 30  
3.3. Value generalization hierarchies ....................................... 32  
3.4. Example of a movement matrix $M$ .................................... 41  
3.5. The measurement model as a weighted directed graph ............. 44  
3.6. Examples of visualizing the probability distribution related to an individual as (a) a flower, and (b) a 'hub and spokes' ....................... 46  
4.1. Measurement approach through Point of Control and Observation (PCO) ........................................... 56  
4.2. Generic approach for using observations at PCOs as input to entropy based privacy measurement ............................... 58  
6.1. Model ................................................................................. 65
List of Tables

1.1. Classification of scale of measurement ............................................. 10
2.1. Basic operations on objects .............................................................. 17
2.2. Microdata table .................................................................................. 24
3.1. Adaptive-interval cloaking algorithm .................................................. 29
3.2. Example: Location information in messages mapped to quadrants ............ 30
3.3. Microdata $T$ ...................................................................................... 33
3.4. Release table $T^*$ .............................................................................. 33
3.5. System microdata $T$ .......................................................................... 33
3.6. Microdata of party $P_1$ .................................................................... 34
3.7. Microdata of party $P_2$ .................................................................... 34
3.8. Anonymous data of party $P_1$ ........................................................... 34
3.9. Anonymous data of party $P_2$ ........................................................... 34
3.10. Join table .......................................................................................... 35
3.11. $T^{**}$: anonymous data of $P_1$ ......................................................... 36
3.12. $T^{**}$: anonymous data of $P_2$ ........................................................ 36
3.13. New join table $T^{**}$ ....................................................................... 36
5.1. Microdata table $T$ ............................................................................. 63
5.2. Release table $T^*$ ............................................................................. 63
5.3. Penalties ............................................................................................ 63
6.1. One tuple without ID attribute ........................................................... 65
6.2. The quasi-identifier for Andy ............................................................... 65
1. Introduction

This deliverable (D2) gives a detailed account of our pioneering work and achievement on the development of measurement approaches for the evaluation and verification of the privacy enabled capability in co-operative Intelligent Transportation System (cITS).

This chapter gives our motivation on the privacy measurement approaches and the relation to other deliverables, followed by an overview on measurement theory and general approaches to privacy measurement. The structure of the rest of the deliverable is given in the end of the chapter.

1.1. Motivation

One of the main objectives of PRECIOSA is to define an approach for privacy evaluation of co-operative systems, with a focus on both communication privacy and data storage privacy\footnote{Note that PRECIOSA objectives precisely address the issues related to data protection explained in the press release from the European commission published on May 2\textsuperscript{nd} 2007.}. This means, in parallel to the specification and implementation of the privacy enhancing mechanisms (PETs) for the privacy-enforceable architecture, we focus on analyzing and developing mechanisms to evaluate and verify the level of privacy provided by the designed system.

This deliverable aims to identify and propose measurement approaches, which will be applied to evaluate and verify cITS privacy at the communication level, at the data storage level, and at the system level\footnote{Notice that in this deliverable, we use the term 'co-operative Intelligent Transportation System (cITS)' . Because the system model considered in PRECIOSA consists of both communication and backend systems (e.g., data storage, service provider). Therefore, we use cITS to refer to the whole system, and the term “V2X communication systems” to refer to the communications part.}. In general, privacy measurements are an important aspect of building trust in co-operative systems, because privacy measurements provide

\begin{itemize}
  \item methods of assessment of trustworthy systems
  \item tools for analysis of vulnerabilities and privacy threats
\end{itemize}

More specifically, privacy measurements of cITS enable us to

\begin{itemize}
  \item estimate the privacy risk for users of cITS
  \item evaluate any privacy protection mechanisms for cITS
  \item verify PRECIOSA privacy-enforceable architecture for cITS
\end{itemize}
1.2. Relation to other deliverables

This deliverable is closely related to other deliverables in the project. Therefore, it is necessary to have an overview of the interrelations of various deliverables to D2.

Figure 1.1 shows the relations among this deliverable and the other deliverables. The main reference of D2 is D7, which specifies the privacy verifiable architecture. D7 decides where the actual privacy measurements will take place in the system. Moreover, the places to measure privacy in the architecture will influence the decision on measurement approaches in D2. D2 also uses the use cases analyzed in D1 to identify possible measurement approaches and their applicabilities. The model and ontology developed in D6 ensure that the measurement approaches in D2 are based on a clearly defined concept and meta model. Some preliminary investigations have been done in D4, which identifies and discusses the challenges and the state-of-the-art of measurement approaches for cITS. D2 continues the discussions in D4 with the focus on the development of concrete privacy measurement approaches and their applications on actual system specified in D7.

On the other hand, D2 also provides input to D7 and D13. The results from D2 will be used to measure the amount of privacy-sensitive information handed through different interfaces in the architecture and to verify the privacy-enforceable features of our design.

![Diagram of relations between deliverables](image)

Figure 1.1.: The relations of D2 to other deliverables

1.3. Short Review on Measurement Theory

Our first step to measure privacy in cITS is to have a better understanding of the theory behind measurement.

In its simplest form, **measurement** is the action of measuring something. According to Wikipedia [2], measurement is *the process of estimating the magnitude of some attribute*
of an object, such as its length, weight, or depth relative to some standard (unit of measurement), such as a meter or a kilogram. Any kind of attributes can be measured. Relatively, metric is a system or standard of measurement, and metrology is the scientific study of measurement.

Obviously, an in-depth discussion of the basic theory behind measurement is out of the scope of this document. However, a short review on general measurement theory is presented and provides a basis for the measurement approaches discussed later in the document.

The basic idea of any measurement theory is that a quantitative scale is a map between some empirical objects and associated numerical values. Suppes and Zinnes [3] identify three problems in the procedure of measurement, which are:

- Justification of the assignment of numbers to objects or phenomena.
- The specification of the degree to which this assignment is unique.
- How measurements may be used, or the meaningfulness of measurement.

To better understand the above three problems, let us consider an example of the measurement of temperature. On the Celsius scale, 0°C is defined as the freezing point of water and 100°C is defined as the boiling point of water. The 0 and 100 here are related to the problem of assigning numbers to the two transition points where water changes its form. The assignment is unique because the empirical meaning of 0°C or 100°C is clear, i.e., they are the degrees of temperature on the Celsius scale. However, imagine a person never heard of Celsius degree before, the sentence “The temperature now is 0°C” will make no sense to him. This is related to the problem of meaningfulness of measurement.

Therefore, when measuring privacy in cITS systems, it is very important that we take the above three issues into consideration. Specifically, any numeric value assigned to reflect the privacy of the system should be justified, and such assignments should be clearly specified to avoid ambiguities. Equally important is the meaningfulness of the measurement. The process of measurement should be well explained and the results from measurements should be interpreted.

Technically speaking, the level of measurement can be categorized into four different kinds of scales [4]. Table 1.1 lists the four scales of measurement.

In nominal scales, numbers are only used as labels. Hence words or letters would serve the same purpose. The numbers assigned in ordinal scales represent the rank order of the objects being assessed. Interval scales are the measurement for “quantitative” attributes of the objects. An arbitrary “zero point” is defined in interval scale. The definition of the zero point is a matter of convention or convenience. Ratio scales can also be used to measure quantitative attributes. To use ratio scales, there must be an absolute zero, even though the zero value may never be produced, e.g., absolute zero temperature.

To apply measurement theory to the measurement approaches in PRECIOSA, we see the possibilities to employ a broad range of measurement scales to estimate and assess
Table 1.1.: Classification of scale of measurement

<table>
<thead>
<tr>
<th>Scale</th>
<th>Basic operation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal</td>
<td>equality</td>
<td>numbering of football players for the identification of the individuals</td>
</tr>
<tr>
<td>Ordinal</td>
<td>greater or less</td>
<td>Evaluation Assurance Levels (1 to 7) given in a Common Criteria evaluation</td>
</tr>
<tr>
<td>Interval</td>
<td>equality of intervals or differences</td>
<td>temperature with the Celsius scale</td>
</tr>
<tr>
<td>Ratio</td>
<td>equality of ratios</td>
<td>temperature in kelvins (relative to absolute zero), or a person’s age</td>
</tr>
</tbody>
</table>

the attributes of cITS and the privacy-enforceable cITS architecture. To our concerns, the attributes are the level of privacy in different aspects (e.g., individual user privacy, user privacy of a specific application, and system privacy etc.) of the system. For example, we can use the ordinal scales to measure whether a privacy-enforceable architecture has 'higher privacy level' than a system without any privacy enforcements. We can also use quantitative measures (in interval and ratio scales) to express that the system can reach a '5.8 degree' of privacy, if 'degree' in the measurement is clearly and meaningfully defined.

1.4. General Approach to Privacy Measurement and Taxonomy

Due to the broadness of the concept, how privacy should be measured is subject to different interpretations. To some, privacy may even appear to be "intangible". To have any meaningful measurement of privacy, it is necessary to identify specific and measurable objects in the general concept and replace the "intangibles" with the clear and understood "tangibles".

Several measurement approaches on privacy have been proposed in the literature. In the following, we will briefly review the most common ones.\(^3\)

**Anonymity.** Pfitzmann et. al [5] define anonymity as the state of being not identifiable within a set of subjects, the anonymity set. The anonymity set is the set of all possible subjects (e.g., a person or a computer). All subjects within the anonymity set have the same attributes (e.g., sending and receiving of particular messages). Anonymity of a particular subject is a concept which is very much context dependent. For example, if a vehicle \( V \) sends a message \( M \) at a location \( L \) and the message does not include any identifiable information, to a potential attacker, all vehicles around \( V \) at \( L \) are possible originators of \( M \). Hence the vehicles, including the sending vehicle \( V \), form an anonymity

\(^3\)The same topic has been discussed in our preliminary investigation in D4
set. \( V \) is said to be "anonymous" within the anonymity set. The size of the anonymity set is measurable and can be used to express the privacy levels of the subjects.

**Unlinkability.** The same authors of [5] define that unlinkability of two or more items of interest (IOIs) (e.g., subjects, messages, and actions etc.) means that within the system (comprising these and possibly other items), from the adversary’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. IOIs are assumed to be the information of interest to an attacker. The attacker tries to relate these items in the system. For example, an attacker may relate a specific message to a specific vehicle. Unlinkability ensures that the probability of those items being related from the attacker’s perspective stays the same before and after the attacker’s observation (attack). The unlinkability can be expressed in probabilities, so it is measurable.

Furthermore, if we consider sending and receiving of messages as the items of interest, we can also define anonymity asunlinkability of an IOI and any subject. Therefore, we can use sender (recipient) anonymity to describe that a particular message is not linkable to any sender (recipient).

In our work, we consider anonymity and unlinkability the primary approaches to privacy measurement. Other notions and measurement approaches exist, e.g., unobservability and pseudonymity. However, they are either derived from these two primary approaches, or not relevant in the context of cITS environment. Notice that D6 includes detailed discussions on various concepts and notions on privacy.

### 1.5. Document organization

Co-operative ITS is a complex system which consists of several conventionally standalone IT systems. Therefore, in Chapter 2, we first identify common measurement requirements and metrics on privacy in general IT systems. To measure privacy values in cITS, we propose specific and detailed measurement approaches in Chapter 3. The actually application of the proposed measurement approaches to the system is discussed in Chapter 4. To have a comprehensive understanding of the system, the overhead incurred by any privacy protection mechanisms and the cost of such mechanisms should also be considered. We call them the complementary aspects to privacy. The measurement approaches to the complementary aspect of privacy are discussed in Chapter 5. Furthermore, Chapter 6 addresses the issue of applying privacy metrics in system development time and runtime. The important issues covered in this deliverable and our conclusion from them are given in Chapter 7.
2. Measurement Requirements and Metrics in IT Systems

There is a diversity of understandings and perspectives on security and privacy, holding by people with different backgrounds. Co-operative ITS is a complex system consisted of a multitude of sub-systems using different Information and Communication Technologies (ICT). To make sure that our measurement approaches are consistent with the requirements and perspectives of the important stakeholders of cITS, in this chapter we investigate the most representative perspectives and requirements on security and privacy in general IT systems. For each of the perspectives, we identify the existing metrics corresponding to their specific requirements.

Therefore, before we propose any specific measurement approaches for cITS, in this chapter we collect requirements on privacy measurement in general IT systems. Since security can be transferred to privacy, e.g., securing data from unauthorized access is one of the prerequisites to protect data privacy, we include both security and privacy in our analysis on measurement requirements.

2.1. Requirements and metrics from Legal and Organizational perspective

Privacy is a fundamental human right, protected by law.

There is a lot of regulations that mandate strong internal controls and protection of Personally Identifiable Information (PII). All of these regulations have common general principles, but could be different or conflicting in details.

Each organization, located in particular country, processing PII is obligated to protect this information according to local law regulations. Moreover, PII could be exchanged with countries having compatible privacy protection laws.

Law regulations compliance is always minimal requirement for system processing PII. PRECIOSA focuses on technical aspects, but protecting privacy is more than technology. Protection of privacy requires a full analysis of information flow in organization, from information acquisition, through information processing to data store. All steps within information processing have to be compliant with privacy protection laws. Moreover, processing PII requires the adaptation of the regulations by the organizations. Privacy protection regulation may require implementing procedures for protecting PII, designating “Privacy
Protection Officer” responsible for protecting PII in organization, and organizing trainings for workers having access to sensitive information etc.

Figure 2.1 gives an overview of the legal and organizational perspective.

![Legal and organizational perspective](image)

Compliance with privacy protection law is a fundamental requirement for systems processing PII and requires full analysis of information flow, organization adaptation and technology solution to create privacy protection environment. Thus the level of compliance is a common measure from a legal and organizational perspective.

### 2.2. Requirement and metrics from system design perspective

This section focuses on the issue of designing and deploying privacy policies as this has an impact on how we can verify that they are enforced. ITS applications are systems which include many embedded systems (e.g. a telematic box in the vehicle, a road side unit etc.). Privacy friendly ITS applications imply that such embedded systems must integrate features that will ensure that privacy policies are followed and enforced. The way these features are integrated into such embedded systems has an impact on the way privacy can be measured, and the way privacy compliance can be verified. As a matter of fact, many embedded systems have resource constraints, and therefore their design is in many case static, i.e. the resources are predetermined at design time. For instance 3 threads, 2 semaphores, 1 serial port etc. While this approach lacks flexibility with respect to dynamic architecture where we can dynamically add a resource, it brings two benefits:

- Systems are more resources aware
- Systems are often more deterministic

A good example of this is the Autosar initiative for automotive ([www.autosar.org](http://www.autosar.org)). In the Autosar engineering approach, engineers configure at build time the resources needed by electronic control unit making up a vehicle. This could involve constants that can no
longer be changed (for instance the priority of a task is no longer in a task control block dynamic memory but a constant in read-only memory (ROM)). This means that there are a number of configuration parameters that are hardwired and unchanged. In PRECIOSA, privacy policies are examples of configuration parameters. A privacy policy could be the following: “erase this data after one hour”. In a static system this policy could be hardwired, meaning that once the system is build up, the policy can never be changed. Note that in many today systems, policies are algorithms or code, and this hardwiring is implicit. The inconvenience is that this is not very flexible (e.g. what if the application designer wishes to change the policy?), but the advantage is that we can trust that the policy is always enforced. We identify the following range of design approaches for the configuration of privacy policies

- OTP capability
  - Approach: in microelectronics OTP (One-Time-Programmable) refers to microcontrollers which can only be programmed once, i.e. code and constants are burned into the microcontroller once (in ROM) at assembly time.
  - Policy deployment: Policies are constants or code which are decided at generation time.
  - Verification of policies: done at design and generation time.

- Maintenance capabilities
  - Approach: policies are considered as stable but they can be changed in a new software release, e.g. when a vehicle is serviced in the garage.
  - Policy deployment: Policies are constants or code which are decided at generation time.
  - Verification of policies: done at design and generation time. At run-time, coherence checking can be applied. While this is not done yet in the vehicle industry we can foresee that mechanisms such as Trusted Platform Modules (TPM) can be used to carry out this checking.

- Runtime capabilities
  - Approach: policies are considered as dynamic constants and can potently be changed dynamically. The impact is that they must be suitably protected and that dynamic modification must rely on specific mechanisms.
  - Policy deployment: dynamic modification through a secure end-to-end channel between the policy manager (PDP – policy decision point) and the electronic (PEP– policy enforcement point).
  - Verification of policies: we need PCOs (Point of Control and Observations) to verify the policies, and we need a secure channel between the verification system and the PCO.

As a result the following requirements are identified

- For OTP capability, verification must be carried out at design and generation time
For maintenance capabilities, verification must be carried out at design and generation time, with some secure mechanism for consistency checking (e.g. based on TPM)

For runtime capabilities, we must rely on a trusted privacy verifiable architecture allowing for secure access to PCOs

The impact on metrics could be the following: the verification level (OTP level, maintenance level, runtime level) could be a parameter used to calculate the privacy level.

### 2.3. Requirements and metrics from Data Security Perspective

Data security is a prerequisite to achieve privacy. The requirements on security such as authentication, integrity, confidentiality, and access control, are also the requirements to build trust in the system and assure privacy. An overview of the related security requirements is given below.

- **Authentication** requires that senders and receivers of any data exchanges are correctly identified.

- **Integrity** requires that any information transported between the sender and receiver should not be altered during transmission, personal information stored in the system should be accurate and up-to-date.

- **Confidentiality** requires that the content of the data exchange between the sender and receiver cannot be understood by a third party.

- **Access control** requires that personal data can only be access by authorized entities in the system.

It is obvious to see that the requirements on data security are transferable to the requirements on privacy. Such requirements are also reflected in the principles of hippocratic databases [6] and hippocratic cITS principles specified in D7.

*Compliance of requirements* is usually used as a metric to evaluate whether a system fulfills the requirements on data security. The evaluation of the level of compliance generally involves the following steps:

- Define a protection profile. A “Security Target” document is created which gives all the information required for the evaluation of compliance of requirements such as assets to be protected, assumptions about the environment, components of the system, counter measures, security requirements, to a class of products through the provision of a “Protection Profile” document. “Protection Profiles” are means to factorize parts of analysis among a category of very similar components. They can be used as they stand (i.e., without any modification), or they can extended with new functional requirements, or augmented with new assurance objectives. In order
to establish confidence in their content and to encourage their reuse, “Protection Profiles” need to be evaluated too. Examples of protection profiles are firewalls, automatic cash dispenser, smart cards, and electronic purses.

- Map the property a system to a set of descriptions or a formal model, e.g., security and access control model like Bell-LaPadula model. Depending on the security target, the mapping process can range from formalization of all descriptions from the high level design to the implementation, and of the correspondence between them, to informal functional descriptions.
- Check the descriptions or formal models against the protection profile, and determine the level of compliance to the security requirements. The resulting level is the compliance to property.

Security is transferable to privacy. To enforce privacy in cITS, the design of the architecture in PRECIOSA employs several security methods. Therefore, level of security becomes relevant to the measurement of privacy level in the overall system. To evaluate a system’s compliance of security requirements, we further investigate and discuss several important issues with major influences on the level of data security in a system. Our focus is on

- security model, in which we discuss the most representative Bell-LaPadula security model;
- industrial perspective, in which we discuss the actual implementation of security models in industry,
- and data security influenced by business activities.

### 2.3.1. Bell-LaPadula model

The Bell-LaPadula model, named after the authors Bell and LaPadula, is a formal security model originally devised for military systems. The key goal is to formalize possible information flows in the system in a way that allows to prove that the system stays in a secure state as long as a strictly defined set of rules is adhered. A first version was published in 1972 [7] which was then amended several times leading to a revised version in 1975 [8]. Several decades later, Bell wrote a paper summarizing all the developments around the Bell-LaPadula model [1].

The most general notion when describing a system are entities. Entities can either be active, i.e., users of the system actively accessing stored data, or they can be passive, i.e., data objects that are accessed by the users. In the following, active entities will be called subjects and passive entities will be called objects.

Subjects can access objects in several ways that either involve the observation of said object’s contents or alteration of the contents. All possible combinations of these abstract concepts give us the well-known basic file operations read, write, append and execute as summarized in Table 2.1. Which subjects can access which objects in which way is
Deliverable 2 v1.0

Table 2.1.: Basic operations on objects.

<table>
<thead>
<tr>
<th>Observation</th>
<th>Alteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>x</td>
</tr>
<tr>
<td>write</td>
<td>x</td>
</tr>
<tr>
<td>append</td>
<td>x</td>
</tr>
<tr>
<td>execute</td>
<td></td>
</tr>
</tbody>
</table>

described in the access matrix $M$. Each cell $M_{ij}$ of $M$ gives the operations that $S_i$ (given by the row) may do on $O_j$ (given by the column).

Further, subjects and objects are assigned a clearance (either unclassified, confidential, secret, or top secret) and a formal category (like “engineers”, “developers”, or “management”). The key property of the clearance is that it can be described by a partial ordering relation. That is, given two clearances $c_1$ and $c_2$ we can say that either $c_1 \geq c_2$ or $c_2 \geq c_1$. This notion of “greater than” will be called domination in the following, i.e. either $c_1$ dominates $c_2$ or the other way round. The tuple of clearance and formal group of a subject is denoted by $f_{S}(S_i)$ and that of an object is denoted $f_{O}(O_i)$. For technical reasons we will also need the clearance that a given subject is currently exerting denoted $f_{C}(S_i)$.

Combining all the previous concepts, we can now formally describe a system state, that is a snapshot of the system at a certain instant in time. The aspects defining such a system state are the following:

1. The Current Access Set describes all current access operations in the system as a set of tuples $(S_i, \alpha_i, O_i)$ where $S_i$ is the subject performing the $\alpha_i$ operation (e.g. read) on $O_i$.

2. The Access Permission Matrix describes the current effective access policies given by a matrix $M$ as described above.

3. The Level Function is given by $f = (f_{S}, f_{C}, f_{O})$ and described the clearances of all objects and the maximum and current clearances of all subjects.

We can now verify the security of any given system state by three concise rules, each of which must hold for every access triple $(S_k, \alpha_k, O_k) \in b$.

- The Simple Security Property is adhered to if $f_{O}(O_k)$ dominates $f_{C}(S_k)$.
- The *-Property is adhered to if
  - $f_{O}(O_k)$ dominates $f_{C}(S_k)$ if $\alpha_k$ is append
  - $f_{O}(O_k)$ equals $f_{C}(S_k)$ if $\alpha_k$ is write
  - $f_{O}(O_k)$ is dominated by $f_{C}(S_k)$ if $\alpha_i$ is read

The 1975 model also envisions to apply a hierarchy upon the objects. This is however not needed for fundamental understanding of the system and therefore omitted in the following description.
The Discretionary Security Property is adhered to if $\alpha_k$ is contained in $M_{ij}$, that is the cell of the access permission matrix describing the operations that $S_k$ is allowed to perform on $O_k$.

A given system state is called a secure state if all properties hold. The simple security property (also called “no-read-up-property”) assures that no subject can observe objects that have a higher clearance than the subject’s maximum clearance. The *-property (also called “no-write-down-property”) restricts all altering operations of a subject to files with the same or a higher clearance. In the original report of Bell this property was missing. That however allowed a malicious subject with a high clearance to read information from secret or even top secret files and write it down to unclassified files thus allowing users with a lower clearance to access secret information as shown in Figure 2.2. To thwart this attack, all users are assigned a current clearance which they may choose freely. This choice however restricts them to write only to files with a higher clearance than their current clearance thus preventing the leaking of secret information.

Note that these first two rules govern access restrictions mandated by the system. It is not at the user’s discretion to change his or her own clearance or that of a file. To allow for more flexibility regarding access restrictions, the last rule was introduced. It allows for users to change the access policies of files owned by them at their liking. All discretionary access rules are however overridden by the mandatory clearance checking which shapes the information flow to fit military regulations.

Given an initial secure state $z_0$ we can now inductively define the notion of a secure system. For that, a set or rules $\omega = \{\rho_1, \rho_2, \ldots, \rho_n\}$ is defined. Each rule addresses a clear change of state, e.g. new file accesses (i.e. changes of $i$) or changes of $M$. Whenever a request $R$ is made in the system, it is matched to a specific rule which governs whether the request is allowed or not. Thus, if it can be verified that each application of a rule to a given secure state results in a secure state, the whole system is secure given that the initial state $z_0$ was secure. The 1975 report of Bell and LaPadula [8] details a set of security property preserving rules for the Multics operation system. For other systems, all possible rules have to be confirmed regarding the three properties in a similar fashion.
This combination of concise rules defining the notion of a secure state and the inductive proof that all possible state transitions of a secure state can only result in a secure state gives a formal backing of a system's security which is very useful, if not required, for military applications.

Application to PRECIOSA

The presented Bell-LaPadula model for mandatory access control is interesting for the PRECIOSA project in two ways. First it establishes a strong formal model to build trust in a system. Although users are allowed to employ discretionary access control, it is always overridden if it is in conflict with mandatory access rules. The second interesting aspect is to take the idea of mandatory access control and transfer the concepts to privacy. That is, it should not be at the service provider’s discretion to change access rules to sensitive information of a user but adherence to policies given by the user should be mandated by the system. This thought is further elaborated in D7.

2.3.2. Industrial perspective on data security

There exists a set of well-established standards in the industry on data storage security. From the industrial point of view, data storage security for privacy compliant system should meet general principles listed below:

- Least privilege – every module (such as a process, a user or a program) must be able to access only those information and resources that are necessary to its legitimate purpose.
- Need to know – Under need-to-know restrictions, even if one has all the necessary official approvals (such as a security clearance) to access certain information, one would not be given access to such information unless one has a specific need to know.
- Data encryption – Privacy protection requires strong data encryption on data storage and transfer.
- Configuration management – Security configuration management ensures persistent privacy protection level.
- Retention of historical data – Privacy protection policies require flexible retention of historical data.
- Audit – Privacy regulations requires logging user actions and events for audit purposes.
- Backup and data recovery – Data availability is one of the basic security attributes.
To implement these principles, industry provides powerful and easy to use set of database tools and features. A set of Oracle security solutions are reviewed in Appendix A as an example of the industrial state-of-the-art.

Within the context of privacy protection, measurements of data storage security need **S**pecific, **M**easurable, **A**ttainable, **R**epeatable and **T**ime-depended (SMART) metrics. Moreover, data storage solutions must meet privacy principles, which have strong influences on data storage of personal identifiable information.

There are a multitude metrics on security, from predominantly technical metrics like the number of denied connections by firewalls, to business-natured metrics like Annualized Loss Expectancy (ALE) or Return Of Investment (ROI).

For privacy protection, applicable security metrics in data storage domain can be:

- **Baseline Defenses Coverage** (Antivirus, Antispyware, Firewall, and so on). This is a measurement of how well the enterprise and data storage system is protected against the most basic information security threats.

- **Patch latency**. This is the time between a patch’s release and the successful deployment of that patch. This is an indicator of a company’s patching discipline and ability to react to exploits, especially in widely distributed companies with many business units. As with basic coverage metrics, patch latency statistics may show machines with lots of missing patches or machines with outdated patches, which might point to the need for centralized patch management or process improvements. At any rate, through accurate patch latency mapping, the proverbial low-hanging fruit (i.e., the easiest targets with the least amount of effort) can be discovered by identifying the machines that might be the most vulnerable to attack.

- **Password strength**. It offers simple risk reduction by sifting out bad passwords and making them harder to break, and finding potential weak spots, e.g., key systems with default passwords

- **Encryption strength**. It can be measured by Time To Break Encryption (TBE).

- **Platform compliance scores**, which use available tools to run tests against systems to find out if the hardware meets best-practice standards. The scoring system is usually simple, and gives a good picture of how the hardware system is "hardened".

### 2.3.3. Multi-level security from a business-oriented perspective

Future ITS applications will be deployed as businesses. It is therefore important to see how these businesses can impact on (1) the PRECIOSA verifiable architecture, and (2) the privacy measurement approaches.

Future ITS infrastructures will allow many independent ITS applications to run in parallel (e.g. tolling, insurance, traffic information etc.). They will likely share some of the computing resources (e.g. a shared telematics box, a shared RSU etc.). This sharing brings the issue of multi-level security. Different sets of possibly conflicting privacy policies might
have to be managed. For instance an application would require the retention a data item while another application would require its elimination. Supporting multiple independent applications has an impact on the architecture and the associated platforms, in particular on the architecture. We identify at least two approaches:

- **Single application Privacy Control Monitor (PCM):** there is one privacy control monitor dedicated per application. Several privacy control monitors can run on top a shared storage and communication system.
- **Multiple application PCM:** the privacy control monitor part of the architecture supports multiple applications.

As a result the following requirements are identified

- **Single application PCM:** verification of proper isolation features in the shared storage and communication system.
- **Multiple application PCM:** verification of proper isolation features in the PCM must also be carried out in addition.

### 2.4. Requirements and metrics from Communications Perspective

The communications perspective focuses on the measurement of communication privacy for V2X communication as well as communication in the infrastructure and back-end domain. Privacy issues in communication systems are discussed in D4. Here our focus is on the requirements and metrics of V2X communications.

#### 2.4.1. Privacy in V2X communications

In many safety applications, vehicles that actively engage in vehicular communications broadcast beacon messages with high frequency to advertise their presence. Although actual beacon rate may depend on application requirements as well as standard specifications, a beacon frequency of 3–10Hz, i.e. one beacon message every 100–300ms [9], is assumed to be realistic. As a result, vehicles almost constantly broadcast and expose verbose information about themselves, because beacons contain information about vehicle location, current speed, heading, and possibly other safety or traffic related information. This constant exposure of vehicle information yields serious privacy issues for vehicles, vehicle owners, and drivers alike. Privacy issues have already been outlined in more detail in D4, together with an overview of privacy protection approaches for V2X systems.

In vehicular communication systems, privacy can either refer to identity privacy or location privacy. Both are issues that have to be addressed. Identity privacy demands that the identity of a vehicle should not be revealed, thus anonymous communication protocols seem to be an adequate choice for V2X communications. The safety critical nature of
some V2X applications, however, requires authentication of valid network nodes to prevent adversaries from posing as vehicles. Furthermore, accountability may also become a legal requirement in the future. As a result, communication systems that provide full anonymity cannot be utilized in V2X systems. Instead, pseudonyms can be used as identifiers to provide anonymous authentication as well as accountability for V2X communication, as is the case in the solution of the European SeVeCom project [10]. In case of legal offenses, authorities would be able to resolve a pseudonym to the corresponding vehicle identity in order to hold the perpetrator accountable.

However, static pseudonyms do not provide adequate location privacy protections, because spatiotemporal information associated with messages or beacons using the same pseudonym can be used to create the location profile of a user, i.e., the patterns and movements of the user of the vehicle. Furthermore, if location information is combined with additional information about these locations the identity of the vehicle or the vehicle owner may be inferred, e.g. if a vehicle is parked at the same house every night it can be assumed that this marks the home location of the vehicle owner. Based on the address the identity of the vehicle owner could be determined, and the pseudonym linked to the identity of the vehicle owner. Therefore, static pseudonyms cannot provide adequate protections to location privacy and identity privacy. As a countermeasure, vehicles should dynamically change their pseudonyms frequently but irregularly to achieve location privacy [11]. However, Schoch et al. [12] show that pseudonym changes with very high frequency disrupt routing and communication functions, therefore, a balance between pseudonym changes and operational requirements is essential.

Pseudonym changes also have to be consistent throughout all communication layers to achieve unlinkability between pseudonyms. When a pseudonym is changed, network identifier and MAC address have to change as well, otherwise linking of subsequent pseudonyms would be trivial. Armknecht et al. [13] propose to derive network identifiers from higher level pseudonyms to take full advantage of their inherent unlinkability.

A different approach for achieving location privacy is the obfuscation or deterioration of location information. While this is a suitable approach for many applications, some safety critical applications, like collaborative collision avoidance, require correct and highly accurate position information to be effective. Therefore, location data distortion cannot be utilized as a general privacy protection approach for V2X communication.

Another important aspect for communication privacy is user empowerment. A user of a communication system should have control about what kind of privacy-sensitive data is communicated to whom and for which purpose [14]. A user can execute this control either by manually consenting or rejecting messages before they are sent out, or by relying on pre-configured privacy policies that do not require user interaction but ensure that communication follows the user’s preferences. The latter are the preferred solution for V2X communication systems, because the communication system should unobtrusively govern the privacy of the driver and the vehicle, instead of requiring interaction of the driver, which would only distract and therefore pose a potential safety hazard.
2.4.2. Privacy metrics in communication domain

As we discussed earlier, privacy in communications can be protected by mechanisms to achieve a certain level of anonymity. A typical measurement for the level of anonymity is the degree of anonymity. The degree of anonymity can be calculated as the size of a user’s anonymity set, i.e. the set of entities that could be the originator of a message or action. Intuitively, a large anonymity set provides better identity privacy, because it is harder for an adversary to successfully determine the actual origin of a message or action.

Location privacy can be measured either as the negated probability of an adversary correctly correlating observed location information from messages with a specific vehicle, or based on the effectiveness of pseudonym changes in terms of tracking. These metrics are entropy-based and do not only take the size of the anonymity set into account but also the probability distribution of the members of the anonymity set. Some entropy-based metrics have been proposed for measuring V2X privacy [15, 16] and will be discussed further in Section 3.3.

In general, in the communication domain, user privacy is mostly protected by mechanisms to achieve either anonymity\(^2\) or unlinkability. Therefore, to measure privacy in communications is to measure the level of anonymity and unlinkability.

2.5. Requirements and metrics from Data Storage Perspective

The data storage perspective concerns the measurement of privacy in each data storage in the system. We mainly focus on the backend domain, where service providers or control units collect, process, and store data. But data can also be stored in the vehicle domain or in the access domain. In the following, we will give a short overview of the main privacy issues and requirement regarding data storage. A detailed discussion can be found in D4.

In the database domain, we assume that data is always stored in a table. Each record or tuple of a table (row) corresponds to one individual with several attributes. We call the original data table microdata and divide the attributes of data tables into three categories [17]. First, attributes that can uniquely identify an individual directly are called *identifiers or key attributes*. Examples are full name, passport number, cell phone number, and social security number. The second category encompasses attributes which in combination can be linked with external information to re-identify (some of) the individuals to whom (some of) the records in the data refer. We call these attributes *quasi-identifiers (QI)*. The last category contains the *sensitive attributes (SA)*. These are attributes which contain sensitive information about the individuals. Examples are medical records, salary, current position, or direction.

\(^2\)The usage of pseudonyms, i.e., to achieve pseudonymity, can be regarded as a special case of anonymity.
A data table is given in Table 2.2. The identifying attributes are SSN and Name, the quasi-identifier are Zipcode and Age, and the sensitive attribute is Current Position.

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Zipcode</th>
<th>Age</th>
<th>Current Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>003</td>
<td>Andy</td>
<td>10121</td>
<td>21</td>
<td>Wall Street</td>
</tr>
<tr>
<td>004</td>
<td>Bob</td>
<td>10122</td>
<td>22</td>
<td>Downing Street</td>
</tr>
<tr>
<td>010</td>
<td>Chris</td>
<td>10123</td>
<td>29</td>
<td>Pennsylvania Avenue</td>
</tr>
<tr>
<td>029</td>
<td>Dave</td>
<td>10401</td>
<td>31</td>
<td>Place de la Concorde</td>
</tr>
<tr>
<td>034</td>
<td>Ellen</td>
<td>10401</td>
<td>31</td>
<td>Atomiumsquare</td>
</tr>
<tr>
<td>059</td>
<td>Frank</td>
<td>10421</td>
<td>34</td>
<td>Unter den Linden</td>
</tr>
<tr>
<td>077</td>
<td>Gillian</td>
<td>10422</td>
<td>39</td>
<td>Münsterplatz</td>
</tr>
</tbody>
</table>

Table 2.2.: Microdata table

In the data storage domain, a privacy breach occurs if an attacker can link sensitive values to specific individuals. For example, in the table above we have to ensure that the attacker is not able to find out, that Andy’s current position is at the Wall Street. So if the table is released the identifying attributes have to be removed before. But the problem is that the combination of the quasi-identifying attributes (Zipcode and Age) can potentially be unique and therefore identifying the individual [17].

Our first approach of privacy measurement will be to measure the anonymity of the individuals in the table after applying mechanisms of anonymizing. On example to enhance privacy is to generalize the values of the quasi-identifier in the table to more general values so that several tuples become indistinguishable. We obtain anonymity sets where we can measure the size. This approach is discussed in chapter 3.2.

The privacy in the storage domain can also be measured with the probabilities that an adversary correctly links sensitive values to individuals. If he knows, that Andy is certainly in the U.S. the probability that he is at the Wall Street (New York) or Pennsylvania Avenue (Washington) is much higher than the probability that he is at the Place de la Concorde (Paris). Such measurement are entropy-based and are presented in chapter 3.3.

2.6. Discussion

In this chapter, we have investigate and identify a set of requirements and metrics with respect to privacy measurement in IT systems. The requirements and metrics are based on viewpoints from various stakeholders, and represent their different perspectives on the topic of privacy in IT systems.

Taking considerations of different perspectives enables us to develop adequate and appropriate measurement approaches understandable to different stakeholders. However, as already mentioned, cITS is a composed system with several sub-systems, which are conventionally standalone. Thus the question is whether we can have a measurement of the overall system, given that we have privacy values of the sub-systems?
To have an overall assessment of privacy in a composed system, it will be helpful to have metrics which integrate different viewpoints. Nevertheless, a metric which integrates all viewpoints is hard to realize due to the inherent differences of the perspectives on privacy among the stakeholders. Despite the differences, based on the common grounds on the technical side, we are able to develop privacy measurement approaches which integrate a part of the viewpoints. The metrics are introduced in the next chapter.
3. Measurement Approaches for Co-operative ITS

This chapter describes the measurement approaches for cITS in details. But first, we introduce the measurement-based verification approach, i.e., using privacy measurements to verify the cITS privacy-enforceable architecture developed in D7.

3.1. Measurement-based Verification Approach

One of the main contributions of PRECiosa is to develop a privacy verifiable architecture for cITS. The framework and various aspects of the architecture are specified in D7. We should be able to give the attestation that the privacy verifiable architecture in PRECiosa complies with the privacy principles. To be able to provide such an attestation, we use design verification and measurement-based verification approaches.

In design verification, the architecture is captured in privacy models, and the design process is verified to see whether it complies with the privacy principles, i.e., the verification of “privacy by design”.

In measurement-based verification approach, V2X privacy measurements provide a rigorous way to verify that the architecture and privacy-protection mechanisms can actually provide a certain level of privacy protection. To measure various privacy values in the system, the information flow among the entities within the system is identified and modeled, based on the high level privacy model in the design. Then the actual measurements are taken at Points of Control and Observation (PCOs) at the information flow.

Although both approaches aim to verify the compliance of the privacy verifiable architecture to the privacy principles, they have different focuses, in which

- design verification focuses on system design. It focuses, for instance, on proving that the designed privacy policies constitute measures that will guarantee the required level of privacy protection.

- measurement based verification approach focuses on system operation, and verifies that the deployed policies conform to the designed privacy policies. The measurement approach depends on the generation approach as explained in Section 2.2:
  - For One-Time-Programmable (OTP) capability, verification is carried out at design and generation time so no specific run-time verification is needed.
For maintenance capabilities, secure mechanism for consistency checking are needed.

For runtime capabilities, measurement features through Points of Control and Observation (PCOs) is needed.

The verification approach is summarized in Figure 3.1.

![Verification approach diagram]

Figure 3.1.: Verification approach

We give detailed description on the selected measurement approaches to privacy values in cITS in the rest of this chapter.

### 3.2. k-Anonymity

#### 3.2.1. Introduction

Pfitzmann and Köhntopp [5] define anonymity as “the state of being not identifiable within a set of subject, the anonymity set”. This definition is further refined with the introduction of the concept of $k$-anonymity by L. Sweeney [17]. A subject is $k$-anonymous if, and only if, it is indistinguishable from at least $k - 1$ other subjects. Thus $k$-anonymity provides a quantitative way to measure and express the degree of anonymity.

---

1PCOs will be elaborated in Chapter 4
3.2.2. Application to communication domain

In the following, we first show how $k$-anonymity has been applied to anonymous usage of location-based services (LBS) in the communication domain. Inspired by the previous work on $k$-anonymity, we propose an approach to apply $k$-anonymity as a privacy metric in the communication domain. We further discuss the pros and cons of $k$-anonymity.

$k$-anonymity applied to anonymous usage of location-based services

Gruteser and Grunwald [18] applied $k$-anonymity in their proposed solution for anonymous usage of users’ location information. In a typical system, mobile users or vehicles regularly send their accurate location information to a location server through cellular or wireless networks. When a user requests a location-based service, the user sends a request to the service provider, which is assumed to be untrustworthy. The service provider obtains the user’s location information from the location server, and tailors its functionalities to the user’s current location. In another scenario, a service provider accesses the location server for vehicle movement data for services like traffic forecasting or road planning.

Therefore, in the system model, the location server acts as a middleware, or a proxy, between a user and the LBS service provider. An anonymity server is implemented as a part of the location server to anonymize the user data, especially location information, before it is passed on to the service provider.

The anonymization is done by an adaptive-interval cloaking algorithm. The desired degree of anonymity is thereby specified by $k$-anonymity. $k$-anonymity is used to measure the degree of anonymity of location information. At most times, a message include temporal information on when it is generated. The temporal information might be either explicitly given in the time stamp of a message or implicitly included as the time the message is recorded. Therefore, location information can be represented by a tuple

\[(x_1, x_2, y_1, y_2, t_1, t_2)\]

in which $[x_1, x_2]$ and $[y_1, y_2]$ define a quadrant in a two-dimensional area where the user is located, and $[t_1, t_2]$ is the time interval during which the user is present in that area. A location tuple of a user is $k$-anonymous, when it describes not only the location of the user, but also the locations of other $k - 1$ users within the time interval given by $[t_1, t_2]$. In other words, $k - 1$ users must be in the area and time period specified by the tuple. The larger the value of $k$, the higher is the degree of anonymity.

The adaptive-interval cloaking algorithm at the anonymity server enforces $k$-anonymity. The desired degree of anonymity in the algorithm is specified by the minimum acceptable size of the anonymity set $k_{\text{min}}$. The algorithm computes an area (quadrant) that includes the actual requester and enough potential requesters to satisfy the anonymity constraint $k_{\text{min}}$. In case of vehicle movement data, the area includes the vehicle from which the

\[Note that such services are very similar to the Floating Car Data use case.\]
location information is generated and enough other vehicles to satisfy $k_{\text{min}}$. The algorithm is listed in Table 3.1 to show how $k_{\text{min}}$ is actually used in the computation. The underlining concept of their algorithm is that a given degree of anonymity can be maintained in any location – regardless of population density – by decreasing the accuracy of the revealed spatial data.

From the above description, we can see that because $k$-anonymity provides a quantitative measurement to describe the degree of anonymity, it enables the development of algorithms to enforce and guarantee user anonymity.

Table 3.1.: Adaptive-interval cloaking algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Adaptive-interval cloaking</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT:</td>
<td>$k_{\text{min}}$, actual position, area covered by anonymity server, position of all other vehicles/subjects in the area</td>
</tr>
<tr>
<td>OUTPUT:</td>
<td>an area (a quadrant) including $k_{\text{min}}$ vehicles/subjects</td>
</tr>
</tbody>
</table>

1: Initialize the quadrants $q$ and $q_{\text{prev}}$ as the total area covered by the anonymizer
2: Initialize a traffic vector with the current positions of all known vehicles
3: Initialize $p$ as the position of the requestor vehicle
4: If the number of vehicles in traffic vector <$ k_{\text{min}}$ then return the previous quadrant $q_{\text{prev}}$
5: Divide $q$ into quadrants of equal size
6: Set $q_{\text{prev}}$ to $q$
7: Set $q$ to the quadrant that includes $p$
8: Remove all vehicles outside $q$ from the traffic vector
9: Repeat from Step 2

$k$-anonymity applied to communication domain

However, our focus here is to use the concept of $k$-anonymity to measure the degree of anonymity of the users involved in V2X communications. Therefore, for a system with $N$ users, $k$ is a positive integer variable, $k = 1, 2, 3, \ldots , N$. It is clear that to reach a certain degree of anonymity, $k$ should be larger or equal to 2, i.e., at least 2 users are indistinguishable from each other with regard to a specific piece of information. If $k$ equals one, the user has no degree of anonymity.

Taking an approach similar to the work of Gruteser and Grunwald [18], we propose the following $k$-anonymity based measurement:

1. Define a time interval $[t_1, t_2]$.
2. Collect all vehicle originated messages, $M = \{m_1, m_2, \ldots , m_n\}$, which pass through one of the points of control and observation (PCO) within the time interval $[t_1, t_2]$. 

04.2009 IST-224201
3. Take the location information contained in the messages and map them to a location tuple \((x_1, x_2, y_1, y_2, t_1, t_2)\), in which \((x_1, x_2, y_1, y_2)\) define a quadrant containing the donor’s (the vehicle’s) location.

4. Calculate the anonymity value \(k\), where \(1 \leq k \leq n\) is the numbers of messages having the same value of \((x_1, x_2, y_1, y_2)\).

5. Use the minimum value of \(k\) as the \(k\)-anonymity value within \([t_1, t_2]\).

**Example** We use a simple example to demonstrate the process to determine the \(k\)-anonymity value. Imagine, within a time interval \([t_1, t_2]\), there are ten messages flowing through one of the PCOs. The location information contained in the message can be mapped to one of the four quadrants, as shown in Table 3.2. The anonymity value \(k\) is then calculated for each quadrant, the result is shown in Figure 3.2. The minimum value of \(k = 2\) is the degree of anonymity at the time interval \([t_1, t_2]\) for the ten messages.

<table>
<thead>
<tr>
<th>(m_1)</th>
<th>(m_2)</th>
<th>(m_3)</th>
<th>(m_4)</th>
<th>(m_5)</th>
<th>(m_6)</th>
<th>(m_7)</th>
<th>(m_8)</th>
<th>(m_9)</th>
<th>(m_{10})</th>
</tr>
</thead>
<tbody>
<tr>
<td>([1,2],[1,2])</td>
<td>([2,3],[1,2])</td>
<td>([1,2],[2,3])</td>
<td>([1,2],[1,2])</td>
<td>([2,3],[1,2])</td>
<td>([2,3],[2,3])</td>
<td>([1,2],[1,2])</td>
<td>([1,2],[2,3])</td>
<td>([1,2],[1,2])</td>
<td>([2,3],[2,3])</td>
</tr>
</tbody>
</table>

**Figure 3.2.** Anonymity value \(k\) at each quadrant.

When applying the \(k\)-anonymity measurement approach to V2X communications, several issues need to be considered.

First, \(k\)-anonymity is a time-variant value, i.e., the values of \(k\) from one of the measurement point will change over time. Consequently, it only reflects the degree of anonymity within a defined period of time.

Second, in the approach proposed above, it is assumed that location information will be part of the message content. However, some V2X applications do not require vehicles...
to include their location information in the messages. Although without explicit location information, these messages might include implicit location information, thus they should be evaluated and included in the $k$-anonymity measurement. For example, consider a roadside unit (RSU) that receives a vehicle generated message by Dedicated Short Range Communications (DSRC) [19] and forwards it to the backend system. If the message contains information unique for the RSU, anyone that knows the exact location of this RSU will also know the approximate location of the sender of the message.

Third, intuitively the smaller the quadrant, the lower the value of $k$ will be. In other words, increasing the accuracy of the location information will decrease the matching of tuples with the same quadrants. This implies that some privacy mechanisms are needed to adjust the accuracy of vehicle-oriented location information if such information will lower the degree of anonymity of the vehicle. Such privacy mechanisms and the architecture (where to place these privacy mechanisms) are given in D13 and D7, respectively. In this deliverable, we will only focus on the measurement of the degree of anonymity.

Discussion

The biggest advantage of using $k$-anonymity is that it is a well-established concept and that the calculation is quite straightforward. Basically, $k$-anonymity is a quantitative way for expressing the size of an anonymity set. The anonymity set is a popular metric in anonymous communication systems. Besides, from the above mentioned algorithm and measurement approach, we can see that the calculation of $k$-anonymity is relatively simple.

However, the simplicity of $k$-anonymity also poses limitations on its capability to measure degrees of anonymity more precisely. Researchers in privacy have already pointed out that the size of the anonymity set does not take into account individual probabilities of each member in the anonymity set with regard to an action [20, 21]. For example, if ten users are probable to be the sender of a particular message, they form an anonymity set with $k = 10$. This calculation is based on the assumption that they are equally probable, so each of them has the probability of $1/10$ to be the sender. Now imagine if one user is more probable to send the message than the others, e.g., this user has sent the message with a probability of $5/10$, while each of the nine others has only $1/18$ probability. Then $k = 10$ dose not accurately reflect the user anonymity level in such situation. In Section 3.3 we will show how to deal with anonymity sets with non-equal probability distributions.

3.2.3. Application to data storage domain

The $k$-anonymity principle is also widely used in the storage domain. As mentioned in Section 2.5 we assume that data is always stored in a table where tuples correspond to individuals and have several attributes. The attributes can be distinguished by identifier, quasi-identifier (QI), and sensitive attributes (SA). Sweeney’s [17][22] concept of $k$-anonymity requires that each tuple of the release table is indistinguishable from at least
$k - 1$ other records with respect to the quasi-identifiers. Let $t[S]$ be the projection of tuple $t$ on the attributes of the set $S$, then we can define the $k$-anonymity property as follows.

**Definition 3.1 ($k$-Anonymity)** A table $T$ satisfies $k$-anonymity if for every tuple $t \in T$ there exist $k - 1$ other tuples $t_1, t_2, \ldots, t_{k-1} \in T$ such that $t_i[Q_I_T] = t_j[Q_I_T] = \cdots = t_{k-1}[Q_I_T]$ for all quasi-identifiers $Q_I_T$.

We call the set of records that share the same values for a quasi-identifier a **QI-group**. $k$-Anonymity means that each QI-group should contain at least $k$ tuples.

**Measurement approach** In D4 we discussed in detail the principle of $k$-anonymity in the storage domain. Here, we present only the approach of measuring privacy with $k$-anonymity. Given a table $T$ with quasi-identifying attributes $A_1, \ldots, A_n$ and a sensitive attribute $S$. The objective is to release a generalized table $T^*$ where we replace the values for the QI with less specific but semantically consistent values. Therefore, for each attribute $A_i$ there is a generalization hierarchy and a level of generalization. $T^*$ is created by a generalization function, which replaces each value of tuples in $T$ with a value in the corresponding level of the hierarchy. Then, we can compute the privacy level $k$ as the minimum size of a QI-group of $T^*$.

**Example** Given the original table $T$ in Table 3.3 and the release table $T^*$ in Table 3.4, which is generated with the help of the generalization hierarchies presented in Figure 3.3. Let Current Position be the sensitive attribute and Zipcode and Age the QI. We want to measure the privacy of the anonymization of $T^*$.

![Value generalization hierarchies](image)

Figure 3.3.: Value generalization hierarchies

In this example we generalize all values to level $1^3$. In Table 3.4 there are 3 QI-groups one of them with size 3 and two with size 2. Therefore, the privacy measure for anonymity is $k = 2$.

---

$^3$See D4 for more details on generalization methods.
3.2.4. Application to system privacy

In the case of system privacy there are several components in the system each of them is $k$-anonymous and the objective is to measure the combination of the data of all components, i.e., privacy values of a composed system.

The simplest example is horizontally partitioned data between two or more system components. In this case all parties have the same attributes but tuples of different individuals. The same holds for the communication domain if different PCOs get messages from different users. We showed in Sections 3.2.2 and 3.2.3 how to measure the anonymity for each component. By merging the information the anonymization criterion is hold because merging in this case means building the union of sets and hence the QI-groups can only become larger.

The more challenging case is the vertically partitioning of data which was analyzed by Jiang and Clifton [23]. The problems occurs if the combination of the data of all components is not anonymous anymore. They introduced an algorithm which can be use to measure the level of anonymization of the join of two data sources before they are joined together. We explain their approach with the help of an example.

<table>
<thead>
<tr>
<th>ID</th>
<th>Zipcode</th>
<th>Age</th>
<th>Salary</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10011</td>
<td>18</td>
<td>80000</td>
<td>Arthritis</td>
</tr>
<tr>
<td>2</td>
<td>10011</td>
<td>19</td>
<td>91000</td>
<td>Cold</td>
</tr>
<tr>
<td>3</td>
<td>10011</td>
<td>19</td>
<td>95000</td>
<td>Heart problem</td>
</tr>
<tr>
<td>4</td>
<td>10021</td>
<td>21</td>
<td>91000</td>
<td>Flu</td>
</tr>
<tr>
<td>5</td>
<td>10021</td>
<td>22</td>
<td>82000</td>
<td>Cancer</td>
</tr>
<tr>
<td>6</td>
<td>10021</td>
<td>23</td>
<td>81000</td>
<td>Diabetes</td>
</tr>
<tr>
<td>7</td>
<td>10022</td>
<td>26</td>
<td>65000</td>
<td>Flu</td>
</tr>
<tr>
<td>8</td>
<td>10022</td>
<td>27</td>
<td>68000</td>
<td>Arthritis</td>
</tr>
<tr>
<td>9</td>
<td>10022</td>
<td>27</td>
<td>69000</td>
<td>Cold</td>
</tr>
<tr>
<td>10</td>
<td>10031</td>
<td>31</td>
<td>53000</td>
<td>Gastritis</td>
</tr>
<tr>
<td>11</td>
<td>10031</td>
<td>32</td>
<td>57000</td>
<td>Flu</td>
</tr>
<tr>
<td>12</td>
<td>10031</td>
<td>33</td>
<td>57000</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>

Table 3.5.: System microdata $T^*$
Let Table 3.5 be the system microdata table, i.e. it contains the merged set of data of all components in the system and Tables 3.6 and 3.7 be the data of party $P_1$ and party $P_2$, respectively. Assume that Disease is the sensitive attribute, Zipcode, Age, and Salary are quasi-identifier and IDs are pseudonyms which identify the rows but not the corresponding individuals. Rows with the same ID value are owned by the same individual. The release tables of parties $P_1$ and $P_2$ are given in Table 3.8 and 3.9.

We can quantify the anonymization of both of the release tables with $k = 3$ which seems to be privacy protected. However, system privacy is not given, because if an adversary knows both of the tables he can easily join them on the ID attribute and obtains Table 3.10 which is only 1-anonymous. The problem is that the QI (Zipcode, Age, Salary) is split over both parties and generalized separately. Thus, the QI-groups of both parties contain different sets of tuples of individuals.
Table 3.10.: Join table

<table>
<thead>
<tr>
<th>ID</th>
<th>Zipcode</th>
<th>Age</th>
<th>Salary</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10011</td>
<td>18–19</td>
<td>80–89k</td>
<td>Arthritis</td>
</tr>
<tr>
<td>2</td>
<td>10011</td>
<td>18–19</td>
<td>90–99k</td>
<td>Cold</td>
</tr>
<tr>
<td>3</td>
<td>10011</td>
<td>18–19</td>
<td>90–99k</td>
<td>Heart problem</td>
</tr>
<tr>
<td>4</td>
<td>10021</td>
<td>20–29</td>
<td>90–99k</td>
<td>Flu</td>
</tr>
<tr>
<td>5</td>
<td>10021</td>
<td>20–29</td>
<td>80–89k</td>
<td>Cancer</td>
</tr>
<tr>
<td>6</td>
<td>10021</td>
<td>20–29</td>
<td>80–89k</td>
<td>Diabetes</td>
</tr>
<tr>
<td>7</td>
<td>10022</td>
<td>20–29</td>
<td>60–69k</td>
<td>Flu</td>
</tr>
<tr>
<td>8</td>
<td>10022</td>
<td>20–29</td>
<td>60–69k</td>
<td>Arthritis</td>
</tr>
<tr>
<td>9</td>
<td>10022</td>
<td>20–29</td>
<td>60–69k</td>
<td>Cold</td>
</tr>
<tr>
<td>10</td>
<td>10031</td>
<td>30–39</td>
<td>50–59k</td>
<td>Gastritis</td>
</tr>
<tr>
<td>11</td>
<td>10031</td>
<td>30–39</td>
<td>50–59k</td>
<td>Flu</td>
</tr>
<tr>
<td>12</td>
<td>10031</td>
<td>30–39</td>
<td>50–59k</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>

The challenge now is the detect and measure this privacy breach and the anonymity of the system without joining the tables. Therefore, for each party $i$, Jiang and Clifton [23] defined a variable $\gamma^*_i$ as a set of sets of ID’s. Each QI-group of each release table form a set of ID’s. In our example there are four QI-groups for $P_1$ (Table 3.8). The first QI-group consists of the first three tuples with ID’s 1, 2, and 3, the second group has ID’s 4, 5, and 6, etc. The release table of $P_2$ has also four QI-group the first with tuple ID’s 1, 5, 6. In summary the variables $\gamma^*_i$ are defined as

$$\gamma^*_i = \{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}, \{10, 11, 12\}\}$$

$$\gamma^*_2 = \{\{1, 5, 6\}, \{2, 3, 4\}, \{7, 8, 9\}, \{10, 11, 12\}\}.$$

The measurement of anonymity based on the equivalence of $\gamma$’s. We say $\gamma_i$ is equivalent to $\gamma_j$, denoted by $\gamma_i \equiv_k \gamma_j$, if there are no $p \in \gamma_i$ and $q \in \gamma_j$ such that $0 < |p \cap q| < k$.

**Theorem 3.1 ([23])** If $\gamma^*_i \equiv_k \gamma^*_j$, then $T^* = T^*_i \bowtie_{ID} T^*_j$ satisfies global $k$-anonymity.

**Measurement approach** In order to measure the anonymity of the whole system we have to compute the values for $\gamma^*$ for each partitioned data set and test for equivalence. Therefore, we compute $k_{\min}$ the minimal size of an intersection of two elements of two different $\gamma^*$’s which is greater than 0. The anonymity $k$ of the system privacy is $k = k_{\min}$.

In the example we can choose $p = \{1, 2, 3\} \in \gamma^*_1$ and $q = \{1, 5, 6\} \in \gamma^*_2$ and obtain $|p \cap q| = 1$. Thus, the joined table $T^* = T^*_1 \bowtie_{ID} T^*_2$ is only 1-anonymous (cf. the first QI-group of Table 3.10).

Jiang and Clifton [23] also presented an approach to obtain a join table which satisfies the previously defined $k$-anonymity constraint. The idea is very simple. If both parties detect the violation of the $k$-anonymity principle if they would join their tables both of them create a more generalized version of their tables. The consequence is that the values for
\(\gamma^*\) change and the sizes of the elements of \(\gamma^*\) increase (because the QI-groups increase). Now, they test the equivalence of the \(\gamma^*\)'s a second time and iterate if the \(k\)-constraint is already violated.

Now, \(\gamma^*\) and the sizes of the elements of \(\gamma^*\) increase (because the QI-groups increase). Now, they test the equivalence of the \(\gamma^*\)'s a second time and iterate if the \(k\)-constraint is already violated.

<table>
<thead>
<tr>
<th>ID</th>
<th>Zipcode</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1001*</td>
<td>18–19</td>
</tr>
<tr>
<td>2</td>
<td>1001*</td>
<td>18–19</td>
</tr>
<tr>
<td>3</td>
<td>1001*</td>
<td>18–19</td>
</tr>
<tr>
<td>4</td>
<td>1002*</td>
<td>20–29</td>
</tr>
<tr>
<td>5</td>
<td>1002*</td>
<td>20–29</td>
</tr>
<tr>
<td>6</td>
<td>1002*</td>
<td>20–29</td>
</tr>
<tr>
<td>7</td>
<td>1002*</td>
<td>20–29</td>
</tr>
<tr>
<td>8</td>
<td>1002*</td>
<td>20–29</td>
</tr>
<tr>
<td>9</td>
<td>1002*</td>
<td>20–29</td>
</tr>
<tr>
<td>10</td>
<td>1003*</td>
<td>30–39</td>
</tr>
<tr>
<td>11</td>
<td>1003*</td>
<td>30–39</td>
</tr>
<tr>
<td>12</td>
<td>1003*</td>
<td>30–39</td>
</tr>
</tbody>
</table>

Table 3.11.: \(T^{**}\): anonymous data of \(P_1\)

<table>
<thead>
<tr>
<th>ID</th>
<th>Salary</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80–99k</td>
<td>Arthritis</td>
</tr>
<tr>
<td>2</td>
<td>80–99k</td>
<td>Cold</td>
</tr>
<tr>
<td>3</td>
<td>80–99k</td>
<td>Heart problem</td>
</tr>
<tr>
<td>4</td>
<td>80–99k</td>
<td>Flu</td>
</tr>
<tr>
<td>5</td>
<td>80–99k</td>
<td>Cancer</td>
</tr>
<tr>
<td>6</td>
<td>80–99k</td>
<td>Diabetes</td>
</tr>
<tr>
<td>7</td>
<td>60–79k</td>
<td>Flu</td>
</tr>
<tr>
<td>8</td>
<td>60–79k</td>
<td>Arthritis</td>
</tr>
<tr>
<td>9</td>
<td>60–79k</td>
<td>Cold</td>
</tr>
<tr>
<td>10</td>
<td>40–59k</td>
<td>Gastritis</td>
</tr>
<tr>
<td>11</td>
<td>40–59k</td>
<td>Flu</td>
</tr>
<tr>
<td>12</td>
<td>40–59k</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>

Table 3.12.: \(T^{**}\): anonymous data of \(P_2\)

<table>
<thead>
<tr>
<th>ID</th>
<th>Zipcode</th>
<th>Age</th>
<th>Salary</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1001*</td>
<td>18–19</td>
<td>80–99k</td>
<td>Arthritis</td>
</tr>
<tr>
<td>2</td>
<td>1001*</td>
<td>18–19</td>
<td>80–99k</td>
<td>Cold</td>
</tr>
<tr>
<td>3</td>
<td>1001*</td>
<td>18–19</td>
<td>80–99k</td>
<td>Heart problem</td>
</tr>
<tr>
<td>4</td>
<td>1002*</td>
<td>20–29</td>
<td>80–99k</td>
<td>Flu</td>
</tr>
<tr>
<td>5</td>
<td>1002*</td>
<td>20–29</td>
<td>80–99k</td>
<td>Cancer</td>
</tr>
<tr>
<td>6</td>
<td>1002*</td>
<td>20–29</td>
<td>80–99k</td>
<td>Diabetes</td>
</tr>
<tr>
<td>7</td>
<td>1002*</td>
<td>20–29</td>
<td>60–79k</td>
<td>Flu</td>
</tr>
<tr>
<td>8</td>
<td>1002*</td>
<td>20–29</td>
<td>60–79k</td>
<td>Arthritis</td>
</tr>
<tr>
<td>9</td>
<td>1002*</td>
<td>20–29</td>
<td>60–79k</td>
<td>Cold</td>
</tr>
<tr>
<td>10</td>
<td>1003*</td>
<td>30–39</td>
<td>40–59k</td>
<td>Gastritis</td>
</tr>
<tr>
<td>11</td>
<td>1003*</td>
<td>30–39</td>
<td>40–59k</td>
<td>Flu</td>
</tr>
<tr>
<td>12</td>
<td>1003*</td>
<td>30–39</td>
<td>40–59k</td>
<td>Diabetes</td>
</tr>
</tbody>
</table>

Table 3.13.: New join table \(T^{**}\)

Example  Party \(P_1\) and \(P_2\) generate their data a second time and obtain Tables 3.11 and 3.12. Then they build the \(\gamma^{**}_i\)'s.

\[
\gamma^{**}_1 = \{\{1, 2, 3\}, \{4, 5, 6, 7, 8, 9\}, \{10, 11, 12\}\} \\
\gamma^{**}_2 = \{\{1, 2, 3, 4, 5, 6\}, \{7, 8, 9\}, \{10, 11, 12\}\}.
\]

Now \(\gamma^{**}_1 \equiv_3 \gamma^{**}_2\) and we can conclude that the join table \(T^{**} = T^{**}_1 \bowtie_{ID} T^{**}_2\) is 3-anonymous. This solution is presented in Table 3.13.
We present another more general measurement approach for system privacy which can be used in order to measure privacy in development time and runtime in Chapter 6.

3.2.5. Interpretation

$k$-anonymity means that a piece of information, in the form of a message or a record, is indistinguishable from at least $k - 1$ other messages or records with respect to a specific data item in the information.

When interpret the values of $k$-anonymity, the following information should also be considered.

- Data item. Based on what specific data item is $k$ calculated.
- Other data items. What are the other data items in the same messages/records.
- Time span and amount of information. Based on how much time and number of messages/records is $k$ calculated.

These information can be regarded as the parameters to describe the meaning of the $k$ value. For example, $k$-anonymity can be calculated on locations or on IDs. Data item describes on which data item (or quasi-identifier in database) is $k$ calculated. In most case, a message or record contains more than one data item. For example, it can contain a user’s ID, vehicle model, and location etc. Maybe $k$ will be high on the data item “vehicle model”, but low on the data item “location”. Therefore, it is necessary to also know the other data items in the same messages/records. Furthermore, without the information on the time span and number of messages/records, $k$ is meaningless. For example, $k$-anonymity on 10 messages in 10 seconds is different from $k$-anonymity on 10000 messages in 1 hour.

For anyone to correctly interpret the measurement result, such information must be included.

3.3. Entropy-based Measures

3.3.1. Introduction

Entropy, or more precisely, information entropy is a quantitative measurement of information content or the uncertainty associated with the information. Entropy was first introduced by C. E. Shannon in his paper on information theory in 1948 [24].

Suppose we have a random process which has a set of possible outcomes with probabilities of $p_1, p_2, \ldots, p_n$. Entropy measures how much “choice” is in the selection of the random process or, in different words, how uncertain we are of the outcome.
Mathematically, the entropy of a random variable $X$ with a probability mass function $p(x)$ is defined as

$$H(X) = - \sum_x p(x) \log_2 p(x)$$  \hspace{1cm} (3.1)$$

The base 2 logarithm is commonly used, so the entropy has a unit of bit, which means how many bits are needed to represent a piece of information.

The basic concept of entropy can be easily understood by an example. Imagine, we have a piece of information that tells us who is the owner of a car. The piece of information reads as:

"The car has 25% chance to belong to Alice, 35% chance to belong to Bob, and 40% chance to belong to Charlie."

Obviously, if we are 100% sure that the car belongs to one person, the information will have no uncertainty. However, we have uncertainty because the information has several possible outcomes. We can use entropy to measure the information content. We have a probability distribution of 0.25, 0.35, 0.4, then the entropy is

$$-0.25 \times \log_2(0.25) - 0.35 \times \log_2(0.35) - 0.4 \times \log_2(0.4) \approx 1.56 \text{ bits}$$

To compare, we can calculate the entropy of being 100% sure as

$$-1 \times \log_2(1) = 0 \text{ bit}$$

which means there is no uncertainty in the information. Entropy reaches its maximum if all elements in the probability distribution are equal, i.e., the maximum uncertainty. For example, if each person is equally probable to be the owner of the car, then the entropy is

$$-\frac{1}{3} \times \log_2\left(\frac{1}{3}\right) - \frac{1}{3} \times \log_2\left(\frac{1}{3}\right) - \frac{1}{3} \times \log_2\left(\frac{1}{3}\right) = 1.59 \text{ bits}$$

The maximum entropy represents an upper bound of the level of uncertainty.

Usually, to protect an individual’s privacy means to protect the personal information from unauthorized access. Various types of unauthorized parties are captured in the adversary model described in D7. For privacy protection, we want information about ourselves to have the highest possible uncertainty to an adversary. Entropy provides the possibility to quantitatively measure the uncertainty associated with the information. Therefore, entropy-based metrics can be used to measure the information in the cITS to decide whether such information is for or against privacy. In general, the higher the entropy, the higher the uncertainty, and thus the higher the privacy value.

There are many ways as how entropy can be used to measure privacy. In the following, we introduce the candidate approaches and discuss them in details.
3.3.2. Application to communication domain

Measurement of anonymity set

Anonymity is an important aspect of privacy and the size of the anonymity set is one of the measurements on the degree of anonymity. If we assume all individuals in an anonymity set have the same probability, we can use the size of the anonymity set to express the degree of anonymity. However, as Diaz et al [20] and Serjantov et al [21] point out, an adversary might obtain additional information in the form of probability distributions of the individuals in an anonymity set. Then the size of anonymity set (e.g., $k$-anonymity discussed in Section 3.2) is no longer an accurate measure of the degree of anonymity.

Based on information theory, authors in [20, 21] propose to use entropy to measure the degree of anonymity. The degree of anonymity is measured as the information an adversary obtained on the anonymity set. In a common scenario, an adversary observes a communication system (e.g., mix networks introduced by D. Chaum [25]) with $N$ users for a while. After the observation, the adversary assigns probabilities to each user in the system as the originator of a message, based on the information the system is leaking, by means of traffic analysis, timing attacks, message length attacks, or more sophisticated attacks.

The maximum degree of anonymity is achieved when the adversary sees all $N$ users as equally probable of sending a certain message. Notice, in this case, we can use the anonymity set size (e.g., $k$-anonymity = $N$) to express the degree of anonymity. However, in most cases, the probability distribution will not be equally distributed. Therefore, entropy is used to quantify the information contained in the probability distribution within the $N$ users, instead of just using $N$ as a measurement.

The degree of anonymity in [20] is calculated as follows. 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, i.e., each $i$ corresponds to an element (a user) of the anonymity set. $H(X)$ is defined as the entropy of the system after the attack. For each user belonging to the set of size $N$, the adversary assigns a probability $p_i$. Entropy $H(X)$ is then calculated as:

$$H(X) = -\sum_{i=1}^{N} p_i \log_2(p_i)$$

The maximum entropy of the system is reached when all $N$ users are equal probable, i.e., $p_i = \frac{1}{N}$. The maximum entropy $H_M$ is

$$H_M = \log_2(N)$$
The information the adversary learned can be then expressed as $H_M - H(X)$. The degree of anonymity of the system $d$ is defined as

$$d = 1 - \frac{H_M - H(X)}{H_M} = \frac{H(X)}{H_M} \tag{3.2}$$

In the calculation of the degree of anonymity, a set of users is defined to belong to the same anonymity set to a particular activity, e.g., sending a message. Although the approach can also be applied to systems with mobility, e.g., vehicle networks, originally it does not take user movement hence location privacy into considerations. In the next section, we introduce approaches extended to cope with location privacy.

**Measurement of mix zones**

Mix zones are geographic areas where an adversary is not able to observe user communications. Typically, mix zones can be achieved by either stopping transmission or by encrypting the communications. The techniques to create mix zones have been discussed in D4.

The concept of mix zones can be applied to a wider area. This means that any geographic area where users’ identifiers or trajectories (i.e., spatial-temporal movements) intersect can be regarded as mix zones. The purpose of mix zones is to mix users to create confusions at the adversary side to provide location privacy protections to the users. Therefore, a whole geographic area can be modeled as a mix zone, the level of location privacy is measured as the level of privacy provided by the mix zone (see D4 for a general introduction on mix zones).

When measuring privacy level of mix zones, Beresford and Stajano [26] point out that anonymity sets do not account for user movement, i.e., they do not model user entry and exit motion of the mix zone. Therefore, they propose to use entropy for a better measurement.

The basic approach is as follows. Consider a user is traveling through a mix zone $z$ at time $t$, we can record the preceding zone $p$ at time $t - 1$, and the subsequent zone $s$ at time $t + 1$. The probability of the pair $(p, s)$ is the probability that a user entering zone $z$ from $p$ and continuing to $s$. Such probability can be statistically determined, e.g., by counting the number of times a person who was in $z$ at $t$ was in $p$ at $t - 1$ and in $s$ at $t + 1$. The pair $(p, s)$ can be stored in a matrix $M$. Imagine we have such a matrix about a 4-way intersection, we might have a matrix like the one shown in Figure 3.4. Based on the matrix, we can then generated the normalized joint probabilities as:

$$p(s, p) = \frac{M(p, s)}{\sum_{i,j} M(i, j)}$$

where $i, j$ are the rows and columns of the matrix, respectively. The conditional probability of a user coming out of zone $s$, having gone in through zone $p$, is calculated as
\[ p(s|p) = \frac{M(p, s)}{\sum_j M(p, j)} \]

The information content associated with a set of possible outcomes with the probabilities \( p_i \) can then be calculated by equation (3.1) to yield an entropy value.

![Figure 3.4.: Example of a movement matrix \( M \)](image)

Buttayán et al [14, 27] further develop the concept of mix zones and apply it to vehicular network. In their approach, a part of the road network is modeled as a mix zone, and level of location privacy provided by the mix zone is measured. It is assumed that the adversary has a model of the mix zone, e.g., the layout of the roads and statistical traffic counts. The mix zone consists of \( n \) gates, through which vehicles drive in and out of the mix zone. The model consists of a matrix \( P = [p_{ij}] \) of size \( n \times n \), and \( n^2 \) discrete probability density functions \( d_{ij}(t) (1 \leq i, j \leq n) \). \( p_{ij} \) is the conditional probability of existing the mix zone at gate \( j \) given that the vehicle enters at gate \( i \). \( d_{ij}(t) \) is the probability distribution of the delay of crossing the mix zone between gate \( i \) and gate \( j \). It is assumed that \( P \) can be obtained by the traffic count, and \( d_{ij}(t) \) can be obtained by traffic flow statistics.

Entropy is used to quantify the probabilities in the model to calculate the level of unlinkability between the entry and exit of vehicles, hence the level of location privacy provided by the mix zone. To calculate the entropy, first an entering event is denoted as \( N = (n, \tau) \), and an exiting event is denoted as \( X = (x, t) \), in which \( n \) and \( x \) are the gates, and \( \tau \) and \( t \) are the time stamps of the events. An adversary make observations of vehicles entering and exiting the mix zone. The set of observed events are denoted as \( (N, X) \). As we can see, each element in the set \( N \) can be mapped to an element in \( X \). However, the adversary does not know the exact one-to-one mapping. Therefore, an adversary makes a set of possible mappings, i.e., to try to link a vehicle exiting the mix zone to the vehicle entering the mix zone. Mathematically, a mapping can be represented by a permutation.
\[ \pi \text{ on } \{1, 2, \ldots, k\}, \text{ in which } k \text{ denotes the } k\text{-th exiting event}^4. \text{ Thus the mapping can be described by the permutation as} \]

\[ m_\pi = (N_1 \rightarrow X_{\pi(1)}, N_2 \rightarrow X_{\pi(2)}, \ldots, N_k \rightarrow X_{\pi(k)}) \]

where \( \pi(i) \) denoted the \( i \)-th element of the permutation \( \pi \), and \( \rightarrow \) denotes the mapping relation. \( \text{Then the conditional probability of the mapping permutation } m_\pi \text{ given the observation } (N, X) \text{ is} \]

\[ p(m_\pi | N, X) = \frac{\prod_{i=1}^{k} p_{n_{x_{\pi(i)}}} d_{n_{x_{\pi(i)}}}(t_{\pi(i)} - \tau_i)}{\sum_{\pi'} p(m_{\pi'}, X | N)} \]

where the nominator is the product of the probabilities given by the matrix \( P \) and the probability density function \( d_{ij}(t) \) with the mapping described by \( m_\pi \). \( \text{The denominator is the sum of products of such probabilities with all possible permutations.} \)

The entropy of the observation \((N, X)\) is then

\[ H(N, X) = -\sum_\pi p(m_\pi | N, X) \log(p(m_\pi | N, X)) \]

So far the measurement is only on a single mix zone. Freudiger et. al [28] sum the entropies of mix zones to express the location privacy achieved by a vehicle \( v \) traversing these mix zones. \( \text{In a vehicle network where several mix zones chain to form a mix-network, the total entropy of the mix-network with } L \text{ mix zones is} \]

\[ H_{tot}(v, L) = \sum_{i=1}^{L} H_i(v) \]

Notice that \( H_{tot} \) is a function of \( v \) and \( L \), meaning that the total entropy depends on the entropies a particular vehicle \( v \) collected at each mix zones.

**A location privacy metric for the users of V2X communication systems**

A metric is a system or standard of measurement. Our goal is to have a metric, which reflects location privacy of the users involved in V2X communications in a quantitative measure. \( \text{Our research into this topic finds out that existing privacy metrics are inappropriate and insufficient to reflect the true underlying privacy value of V2X communication systems.} \)

\[ \text{Therefore, we have developed a novel privacy metric within the PRECIOSA} \]

\[ ^4\text{Notice that } k \text{ here is an index, which has different meaning as in } k\text{-anonymity} \]
Deliverable 2 v1.0

Our analysis shows that in the context of V2X communications, the properties of location privacy consist of individuals and their trip information. Therefore, we can use an adversary's ability to link vehicle trips to specific individuals to reflect the level of location privacy of the individuals.

A V2X communication system is a dynamic system and continuous in space and time. To allow us to take a sensible measurement of the system, we need to take a discrete sample from the system and base our measurement on a relatively static and confined version. Thus we make three assumptions:

1. The location information considered in the metric is assumed to be within an arbitrarily defined area.
2. The location information considered in the metric is assumed to be within an arbitrarily defined time period.
3. We further assume that the adversary is able to identify a location as the origin or destination of a trip.

Combining these three assumptions, we derive that there will be only complete trips and the number of origins and destinations are equal. The first two assumptions enable us to virtually take a snapshot of the system. The snapshot captures the vehicle movements and their relations to the drivers in a given area and time period.

To quantitatively measure user location privacy, we take the following steps. In the first step, we model the information contained in the snapshot in a measurement model. The measurement model abstracts the location information into three basic components: the linkage of an individual to an origin of a trip, the linkage of an origin to a destination, and the linkage of a destination to an individual. The adversary’s knowledge of the system is expressed as probability assignments on each of the linkages. In the second step, the probability distributions in the measurement model are extracted to yield quantitative measurements. The two steps are described in details in the following sections.

Modeling the information contained in the snapshot is to represent the information in an abstract and mathematical form to facilitate calculation in the next step. We observe that the information in the snapshot contains the information on individuals, O/D pairs, and their interrelations. We also observe that for an individual to ‘make a trip’, he or she must start a trip at an origin and ends the trip at a destination. This also implies that the individual at the origin and the destination should be the same person.

Based on the observations, we model the information as a weighted directed graph $G = (V, E, p)$ shown in Figure 3.5. There are three disjoint sets of vertices in the digraph, i.e., $I \subseteq V$, $O \subseteq V$, and $D \subseteq V$ with $I \cup O \cup D = V$. $I$ is the set of all individuals, $|I| = n$. $O$ is the set of all origins and $D$ is the set of all destinations of the trips, $|O| = |D| = m$. The edge set $E$ is defined as $E := E_1 \cup E_2 \cup E_3$ with $E_1 := \{e_{io}|i \in I, o \in O\}$, $E_2 := \{e_{od}|o \in O, d \in D\}$ and $E_3 := \{e_{di}|d \in D, i \in I\}$. As $E_1, E_2, E_3$ are disjoint, $G$ is a tripartite graph. Each edge $e_{jk} \in E$ is weighted by a probability function $p : E \rightarrow [0, 1]$. 
$G$ has several notable properties. First, $G$ contains all aforementioned information in the snapshot. Since tracking is not the focus of the paper, we assume that there is a publicly known tracking algorithm and treat vehicle tracking as a black box, i.e. we assume that $p(o_j, d_k)$ is known. Second, vertices in $G$ are connected with directed edges. If we follow the directed edges from a vertex $i_x$, the path will pass the vertices $\{i_x, o_j, d_k, i_y\}$. The semantics of the cycle is $i_x$’s possibility having made a trip from $o_j$ to $d_k$. Third, the probability distributions on the edges model an adversary’s knowledge of the users and their movements in the system. In addition, we define that the sum of the probabilities on outgoing edges from a vertex $o \in O$ or $d \in D$ to be $1$, $\sum_k p(o_j, d_k) = 1$, $\sum_k p(d_j, i_k) = 1$, while letting the sum of probabilities from the vertex $i \in I$ to be equal of smaller than $1$, $\sum_k p(i_j, o_k) \leq 1$. By the latter definition, we model an individual not making any trips. For example, $\sum_k p(i_1, o_k) = 0.9$ means that $i_1$ has 0.9 probability to make trips and 0.1 probability to ‘stay at home’.

For the ease of calculations, we also represent $G$ by three adjacency matrices, $IO$, $OD$, and $DI$.

\[
IO = \begin{bmatrix}
p(i_1, o_1) & p(i_1, o_2) & \cdots & p(i_1, o_m) \\
p(i_2, o_1) & p(i_2, o_2) & \cdots & p(i_2, o_m) \\
\cdots & \cdots & \cdots & \cdots \\
p(i_n, o_1) & p(i_n, o_2) & \cdots & p(i_n, o_m)
\end{bmatrix}
\]

\[
OD = \begin{bmatrix}
p(o_1, d_1) & p(o_1, d_2) & \cdots & p(o_1, d_m) \\
p(o_2, d_1) & p(o_2, d_2) & \cdots & p(o_2, d_m) \\
\cdots & \cdots & \cdots & \cdots \\
p(o_m, d_1) & p(o_m, d_2) & \cdots & p(o_m, d_m)
\end{bmatrix}
\]

\[
DI = \begin{bmatrix}
p(d_1, i_1) & p(d_1, i_2) & \cdots & p(d_1, i_n) \\
p(d_2, i_1) & p(d_2, i_2) & \cdots & p(d_2, i_n) \\
\cdots & \cdots & \cdots & \cdots \\
p(d_m, i_1) & p(d_m, i_2) & \cdots & p(d_m, i_n)
\end{bmatrix}
\]

Each entry $a_{jk}$ in the matrices indicates that there is an edge from vertex $v_j$ to vertex $v_k$. The value of the entry is the weight on the edge, $a_{jk} = p(v_j, v_k)$. Furthermore, each row in the matrices is a vector of the probability distribution on all outgoing edges from the same vertex.
Deliverable 2 v1.0

vertex. The sum of each row in $\text{IO}$ is equal or smaller than 1, and the sum or each row in $\text{OD}$ and $\text{DI}$ equals 1.

To extract the probability distributions and quantify the information in the measurement model, we use information entropy developed by Shannon [24]. Entropy is a quantitative measure of information content and uncertainty over a probability distribution. Entropy has been widely accepted as an appropriate measure in the privacy research community [21, 20, 26]. However, the main challenge here is to apply entropy to the measurement model.

By definition, for a probability distribution with values $p_1, \ldots, p_n$, the entropy is

$$ H = - \sum p_i \log p_i $$

where $p_i$ is the $i^{th}$ element of the probability distribution. $H$ is then a measure of information content and uncertainty associated with the probability distribution. The logarithm in the formula is usually taken to base 2 to have a unit of bit, indicating how many bits are needed to represent a piece of information. A higher entropy means more uncertainty and hence a higher level of privacy. Entropy reaches its maximum value if all the probabilities in the distribution are equal.

Shannon uses entropy as a quantitative measure of the information produced by a discrete information source. When applying entropy to our calculation, the source is the information captured in the measurement model accessible to the adversary. We are interested in the information on the relations between the individuals and the trips, i.e., the information on who moves from where to where. The information is expressed as the probabilities of a particular individual within the system to make one of the trips, as well as to not make any trips.

We are interested in the entropy (the uncertainty) related to an individual and the $m$ O/D pairs (which leads to $m^2$ possible trips) in the system. If we 'unfold' all the cycles related to a particular individual in $G$ (cf. Figure 3.5), we obtain a flower-like structure shown in Figure 3.6(a). The stigma, or the center of the flower is the individual, e.g., $i_1$. The petals run clock-wise around the stigma, denoting $i_1$ making one of the $m^2$ possible trips, with the last petal representing $i_1$ does not make a trip. We denote this complementary probability $p^c$. If we assume that the measurements reflect separate observations, i.e., the probabilities describe independent events, the probability of an individual making a specific trip is the product of the probabilities on all edges of the petal representing that trip. We can further simplify the flower structure to the wheel-like structure in Figure 3.6(b). The hub in the center represents an individual, e.g., $i_1$. Each radiating spoke from the hub represents the probability of $i_1$ making a specific trip. In other words, it shows how much chance $i_1$ has taken her or his car and driven from a location $o_j$ to another location $d_k$. The last spoke represents the situation that $i_1$ has taken none of the trips, i.e., $p^c$ is the probability that $i_1$ stays 'at home'.

We take the non-zero probabilities and normalize them to calculate the entropy, because $p_i = 0$ means there is no uncertainty and the sum of the probability distribution should
equal 1. Based on Formula (3.4) and using the notation specified in the measurement model, we calculate the entropy for a specific individual as

\[ H(i_s) = -\left( \sum_{j=1}^{m} \sum_{k=1}^{m} \hat{p}_{jk} \log(\hat{p}_{jk}) + \hat{p}^c \log(\hat{p}^c) \right) \]  

(3.5)

where \( \hat{p}_{jk} \) is the normalized probability of \( i_s \) making a trip from \( o_j \) to \( d_k \) and \( \hat{p}^c \) is the normalized probability of \( i_s \) not making any trips. The values of \( \hat{p}_{jk} \) and \( \hat{p}^c \) are given as

\[ \hat{p}_{jk} = \frac{p(i_s, o_j)p(o_j, d_k)p(d_k, i_s)}{\sum_{j=1}^{m} \sum_{k=1}^{m} p(i_s, o_j)p(o_j, d_k)p(d_k, i_s) + \hat{p}^c} \]  

(3.6)

in which

\[ \hat{p}^c = 1 - \sum_{j=1}^{m} p(i_s, o_j) \]  

(3.7)

To evaluate the location privacy of an individual, it is also useful to find the maximum entropy possible for an individual in the system, i.e., the upper bound. The maximum entropy for an individual is reached if all of the participants in the system are equiprobable to make any trips and all trips are also equiprobable. In a system with measurements of \( m \) O/D pairs, the maximum entropy of an individual \( i_s \) is

\[ MaxH(i_s) = -\log\left( \frac{1}{m^2 + 1} \right) \]  

(3.8)

where 1 in the denominator accounts for the individual not making any trips. Interestingly, the maximum entropy for an individual depends only on the number of possible trips, not the number of participants in the system.

Given the entropy upper bound, the level of location privacy of an individual can also be expressed as the ratio of the current entropy to the maximum. Therefore, we have

\[ H\% = \frac{H(i_s)}{MaxH(i_s)} \times 100\% \]  

(3.9)
which uses $\%$ as the unit. We use $H_{\%}$ to express the ratio of an individual’s privacy level to the maximum possible level. In other words, it gives a hint as how far an individual is from the theoretical privacy upper bound.

### 3.3.3. Application to data storage domain

The concept of entropy-based measurement can easily be applied to the storage domain. As mentioned in Section 2.5 we regard tuples in a data set belong to individuals. Each tuple has at least one sensitive attribute value. Consider a random process which links the individuals to sensitive values. Let $S$ be the sensitive attribute with possible values $s_1, s_2, \ldots, s_n$ and $p_i$ be the (linking) probability that the sensitive value of a given individual is $s_i$. Then, uncertainty about $S$ can be measured as the entropy of $S$.

$$H(S) = - \sum_{i=1}^{n} p_i \log_2 p_i$$  \hspace{1cm} (3.10)

Notice that this formula is similar to Equation 3.1 with the random variable $S$. We can measure the privacy of a data storage and the information an adversary learns by comparing several uncertainties. Therefore, we need a reference value of uncertainty. In Section 3.3.2 we introduced the maximum entropy $H_M$ and take the difference between $H_M$ and $H(S)$ for the measure of information gain. The intuition behind this idea is that it is desirable that an adversary has the maximal uncertainty about the sensitive values, i.e. the same linking probabilities. In the data storage domain this is not always preferable. Consider for example a sensitive value Virus test with values Positive and Negative. Let 5% of all tuples in the data set share the sensitive value Positive, i.e. they are afflicted with the virus and 95% have value Negative. The data is anonymized and released. Then, if an adversary can conclude that there are only 10 tuples which can be linked to a certain individual and 5 of them have sensitive value Positive and 5 have Negative his linking probabilities changes to 0.5 for both values. His uncertainty is $H(S) = -(0.5 \cdot \log 0.5 + 0.5 \cdot \log 0.5) = \log 2 = 1$ which is the maximal value for $H(S)$ but his information gain is enormous. The adversary can conclude that a specific individual has a 50% chance of being afflicted with the virus which is a 10 times higher probability than his knowledge before the data release. If we use the maximum uncertainty as reference value we measure the degree of anonymity with Equation 3.2 as $\frac{H(S)}{H_M} = 1$ which does not reflect the information gain in this case. Hence, we need a more meaningful reference value in order to be compared with the uncertainty of a given release.

**Measurement approach** In order to measure the privacy we define a reference entropy $H(S)$ which reflects the uncertainty of an potential adversary before an attack or an information gain. If we assume no background knowledge the reference entropy is the maximum entropy $H_{\max}$ otherwise it is computed with Equation 3.10. Now suppose the adversary gets information about the distribution of the sensitive attribute by an attack or
a release of a table. His probabilities of sensitive values change to $p'_i$ and his uncertainty about $S$ to

$$H(S') = - \sum_{i=1}^{n} p'_i \log_2 p'_i.$$ 

We measure the privacy of the data by comparing his old knowledge $H(S)$ with his new knowledge $H(S')$. There are several measures proposed in literature. Privacy measures are

- the degree of anonymity: $\frac{H(S')}{H(S)}$,
- the loss of privacy [30]: $1 - 2^{-(H(S) - H(S'))}$,
- the Kullback-Leibler divergence [31]: $KL(S, S') = \sum_{i=1}^{n} p_i \log \frac{p_i}{p'_i}$.

The degree of anonymity computes the anonymity in the new distribution related to the knowledge before. The values are between 0 and 1. A degree of 0 means that there is no anonymity which ensures $H(S) = 0$ and the knowledge that there is only one possible value for $S$. The maximum value is reached if $H(S) = H(S')$.\(^5\)

A user’s loss of privacy is directly related to the amount of information gained by an attacker from the attack. The value range is $[0; 1)$. In the case of $H(S) = H(S')$ the loss of privacy is $1 - 2^0 = 0$ and in the case of $H(S') = 0$ (only one possible value) it is $1 - 2^{-H(S)}$ which can be arbitrary closed to 1.

The last measurement presented here is the Kullback-Leibler divergence. It is also called relative entropy and computes the different between two probability distributions. KL measures the expected number of extra bits required to code samples from $S$ when using a code based on $S'$, rather than using a code based on $S$. This measurement is always greater or equal to 0 and it can be shown that $KL(S, S') = 0 \iff S = S'$. Although it is often intuited as a distance metric, the KL divergence is not a true metric since it is not symmetric and does not satisfy the triangle inequality.\(^6\)

<table>
<thead>
<tr>
<th>Zipcode</th>
<th>Age</th>
<th>Current Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1012*</td>
<td>20–29</td>
<td>USA</td>
</tr>
<tr>
<td>1012*</td>
<td>20–29</td>
<td>United Kingdom</td>
</tr>
<tr>
<td>1012*</td>
<td>20–29</td>
<td>USA</td>
</tr>
<tr>
<td>1040*</td>
<td>30–39</td>
<td>France</td>
</tr>
<tr>
<td>1040*</td>
<td>30–39</td>
<td>Belgium</td>
</tr>
<tr>
<td>1042*</td>
<td>30–39</td>
<td>Germany</td>
</tr>
<tr>
<td>1042*</td>
<td>30–39</td>
<td>Germany</td>
</tr>
</tbody>
</table>

Table 3.14.: New release table $T^*$

---

\(^5\)We always assume $H(S') \leq H(S)$

\(^6\)Hence, we call it 'divergence' rather than 'distance'.
Example  In the storage domain the knowledge of the distribution of the sensitive attribute is often assumed to be known by the adversary. Recall Table 3.3 and assume an adversary knows all the sensitive values of attribute Current Position: Wall Street, Downing Street, Pennsylvania Avenue, Place de la Concorde, Atomiumsquare, Unter den Linden Münsterplatz. Imagine, he is interested to know in which countries the six individual are. The Wall Street and Pennsylvania Avenue are in the USA, Downing Street is in the United Kingdom, Place de la Concorde in France, Atomiumsquare in Belgium and Unter den Linden and Münsterplatz are in Germany. His knowledge can be measured with the entropy of the sensitive attribute Current Country as reference value $H(S) = H(\text{Current Country})$.

$$H(\text{Current Country}) = - \left( \frac{2}{7} \cdot \log_2 \left( \frac{2}{7} \right) \cdot 2 + \frac{1}{7} \cdot \log_2 \left( \frac{1}{7} \right) \cdot 3 \right)$$

$$= 2.236$$

Then the adversary sees the release Table 3.14 and his uncertainty changes. For example, if he knows Andy’s age 21 and his zipcode 10121 he concludes that Andy’s tuple is in the first QI-group and has possibles values of USA or United Kingdom. That means that there are only two countries left where Andy could be. His new uncertainty about Andy is

$$H(\text{Current Country}') = - \left( \frac{2}{3} \cdot \log_2 \left( \frac{2}{3} \right) + \frac{1}{3} \cdot \log_2 \left( \frac{1}{3} \right) \right)$$

$$= 0.918$$

because the value USA occurs twice in the first QI-group. If the adversary further knows Frank’s QI values (34 years old, zipcode 10421) he concludes that Frank is in Germany. The entropy of Frank’s sensitive value is

$$H(\text{Current Country}') = -(1 \cdot \log_2 1) = 0.$$  

The degree of privacy for Andy is

$$d = \frac{H(S')}{H(S)} = \frac{0.918}{2.236} = 0.411$$

and for Frank

$$d = \frac{0}{2.236} = 0$$

which captures the intuition of the privacy breach of Frank’s sensitive value Current Country.
\textbf{l-Diversity}

Another interesting entropy-based measurement is the principle of entropy \( l \)-diversity which was introduced by Machanavajjhala et al. [32]. Their objective is to extent the \( k \)-anonymity principle and ensure that in all QI-groups there are multiple values for the sensitive attribute. A QI-group is \( l \)-diverse if it contains at least \( l \) "well-represented" values for the sensitive attribute. A table is \( l \)-diverse if every QI-group is \( l \)-diverse. On interpretation of this principle is called entropy \( l \)-diversity and claims that the entropy of the sensitive attribute values distribution in each QI-group can be bound by an constant \( \log l \).

\[ - \sum_{i=1}^{n} p_i \cdot \log p_i \geq \log l \]

**Example**  The first QI-group of Table 3.14 is entropy 1.889-divers: \( -(\frac{2}{3} \cdot \log \frac{2}{3} + \frac{1}{3} \cdot \log \frac{1}{3}) \approx 0.918 \leftrightarrow l \leq 2^{0.918} \approx 1.889. \) The second group is 2-divers and the last group is 1-diverse.

We realize that 1-diversity does not protect privacy. From the properties of the logarithm function follows that in order to have entropy \( l \)-diversity for each QI-group, the entropy of the entire table must be at least \( \log l \). Furthermore, each entropy \( l \)-diverse table satisfies \( l \)-anonymity.

\subsection{3.3.4. Application to system privacy}

Entropy-base measurements are very general. The idea of measuring the entropy before an action (attack, release, \ldots), after an action and comparing these entropy values can easily be adapted in the whole system. In the same way, we can also compare the entropy values of cITS with and without any privacy-protection mechanisms to have quantitative measures of privacy achieved by the mechanisms.

Another features of entropy is that it is a quantitative measurement of information content. Privacy in cITS is about how information of the users are collected, transported, processed, used, and stored. Therefore, entropy is applicable to all parts of the system, where information is of concern.

\subsection{3.3.5. Interpretation}

Entropy is a measure of information content and uncertainty. However, entropy itself does not explicitly specify the type of information it quantifies. Therefore, when presenting the result of entropy-based measurement, a description of the information, on which entropy is calculated is necessary. The information can be divided into two coarse-grained categories:

- Information on anonymity. For example, the entropy of anonymity set.
• Information on (un)linkability. For example, entropy of linkability of vehicle traversing mix zones.

Beside the fact that entropy should always be interpreted together with the information it quantifies, the values of entropy only make sense when they can be compared to reference values, e.g., the upper bound, the entropy before or after an attack, or the entropy with or without a protection mechanism.

### 3.4. Metrics on Data Security

As mentioned in Section 2.3, data security is transferable to privacy. Therefore, in addition to the metrics on privacy discussed so far, we also investigate the possible measurement approaches to data security.

Unlike most research-oriented privacy metrics, there are a set of well-established security metrics in research and industry. In this section, we focus on the approaches to the quantification of the level of protection of data against attacks and abuse while being stored, transmitted and backed-up. We based our discussion on a set of Oracle white papers.

Oracle provides a set of products and options to secure the information which assists in meeting various governement and industry requiriements\(^7\), the Oracle Maximum Security Architecture.

The level of protection is dependent of which and up to which level the different parts are installed and configured. It is not sufficient to implement one or a few portions of the set up to a very high level while ignoring other parts. The maximum security architecture has several sections: user management, access control, encryption and masking and monitoring.

Below is a short description of different parts of the aspects of data protection practices. Maximum security is achieved when all applicable options are implemented correctly and with great care. Quantification of the strength of the data protection is not simply adding up the number of adopted items it the list below. Furhtermore failing to setup one of the options might make others less efficient.

#### 3.4.1. User Management

**secure passwords**: implementation of a strong password policy and forcing users to choose secure password is the first and one of the most important protection against connection level threats.\(^{[33]}\)

\(^7\) SOX, PCI, HIPAA, FISMA, JSOX, EU DATA PRIVACY DIRECTIVE
centralised user management: managing database users and authorisations in one central place improves maintenance of security procedures significantly. See [34]. Oracle database security/enterprise use security.

strong authentication: two-factor (or “strong” authentication is based on something the user has (a smart card, token, etc…) and something the user knows (a PIN or passcode). Examples of industry–standard authentication methods are Kerberos, RADIUS, SSL or Secure Sockets Layer, PKI or Public Key Infrastructure. See also [35].

proxy authentication for securing access to multi-tier applications, when the identity of the end user is required at database side. See [34] and [36].

White paper [37] contains a checklist with respect to access to stored data.

3.4.2. Access Control

privileged user controls to prevent e. gd. users with DBA role to access data with respect to strong internal control mandates found in regulations such as Sarbanes-Oxley, insider threat concerns, protection of personally identifiable information (PII), IT or DBA outsourcing. See [38] for more on Oracle Database Vault.

controlling who, when, where, how data and applications are accessed: due to insider threat concerns, protection of personal identifiable information and legal regulations, it is important to limit access to database/datawarehouses to approved applications. This can be achieved using command rules and multi-factor authorisation. See also [39].

row and column level security or fine grained access control, allows to creates data access policies when object privileges and database roles are not sufficient. The Oracle virtual private database provides a variety of option within this context. See also [40].

multi–level security permits access by users with different security clearances and needs–to–know, and prevent users from obtaining access to information for which they lack authorization. See also [41] and [42] on multi–level and label security.

data classification to control access to information. This option is closely related to the previous one. See also [42] and [39].

3.4.3. Encryption and Masking

data encryption: transparant data encryption encrypts/decrypts data when it is written/read, without interference of applications and as such inhibits data stored as plain text. Oracle supports also network based Hardware Security Modules. See [35] and [43].
**network encryption**: data is especially vulnerable when being transmitted over the network. To protect integrity, data needs to be encrypted and signed. See also [35].

**data masking** is needed when production/confidential data is copied or shared for example application testing. Replacing sensitive data irreversible with realistic looking information is required here. For more information, see [44].

**export encryption**: exports are logical backups of data intended to move relatively small amount of dat around. Care has to be taken this information is protected by encryption. See [45].

**backup encryption**: physical backups to tapes or disks have to be secured using strong encryption. See [46]. Encryption methods can be based on pass phrases or on encryption keys([43]).

### 3.4.4. Monitoring

When connection policies, data access policies and data protection policies are in place, those have to be monitored carefully, to be able to take appropriate actions in case of abuse (attempts) and to adjust policies eventually.

**database auditing** or keeping track of who is performing what operation when on what data is the basic information needed. [47] and [48].

**fine grained auditing** or policy-based auditing provides more meaningful audit data, as only information is produced at specified conditions. See [49] and [48].

**audit consolidation, reporting and alerts**: the collected audit data is useless if not analysed and appropriate actions taken. Oracle Audit Vault automates the collection of audit data from a variety of database environments such as Oracle, SQL Server, IBM DB2 and Sybase ASE. See also [48].

**secure configuration scanning**: in general, IT infrastructures are continuously changing and evolving, components are added, removed, functions are changed. Policies and applications for continuously monitoring the configuration are needed to reduce the risk of security issues. See [50].

### 3.5. Discussion

As mentioned in Section 2.6, we are interested in metrics which can represent a broad range of perspectives. In this sense, $k$-anonymity can be used in both communication and data storage domain to represent the degree of anonymity. Technically, the main operations on privacy-relevant data in cITS are data communication, data storage, and data usage. $k$-anonymity can provide measurements on how the data is communicated in the system and accessed in the storage, thus help to monitor the operations of privacy-relevant data in the system.
It should be noticed that when $k$-anonymity is used for the whole system, the data item of which $k$-anonymity is measured should be consistent. For example, if we use $k$-anonymity to express the anonymity level of a user’s location in the communications, then $k$-anonymity in data storage should also measure the user’s location, not the user’s identifier.

Entropy is an established and well-recognized measurement of information content and certainty. It is also an appropriate measurement to reflect privacy values in cITS, where sensitive information such as identities and locations is of concern. Entropy is applicable to both communication and data storage domain. This gives us the advantage of having another cross-domain measurement in the system.

Same as $k$-anonymity, when applying entropy-based measurement to the system, it is important to have a consistent information model on which entropy is calculated. For example, if we calculate entropy of the linkability of location information to specific individuals in communication domain, the semantics of entropy in data storage domain should be the same, i.e., the linkability of the attributes of locations to the attributes of identifiers in the data table.

Security contributes to the privacy level of the system. Thus metrics on security provide an additional source of measurements which can be used to verify and monitor the privacy protections throughout the system. However, the measurements on security are orthogonal to the measurements on privacy. Nevertheless, the list of security metrics in Section 3.4 can serve as a security checklist when evaluating the privacy-protection of the system and provide the level of assurance on data protection in the system.
4. Measurement through PCOs in Information Flow Analysis

4.1. Methodology

By information flow analysis, we mean to keep tracks of personal and privacy-sensitive information that is transmitted and processed by entities in cooperative ITS. The main goal of the analysis is to identify the points of control and observation (PCO) in the system, where any personal and privacy-sensitive information is exposed to potential privacy attacks. By taking measurements using one of the aforementioned approaches, we can qualitatively and quantitatively analyze and assess the information leaks and the extend of the privacy protection of the privacy verifiable architecture developed in PRECIOUSA. The process to identify POCs has been described in details in D7.

Identifying PCOs for information flow analysis involves the following two steps:

1. Identify Points of Attack (PA) in the system through an adversary model.
2. Identify PCOs at or around PAs in the system.

PCOs are a set of points, at which the measurement approaches can be applied to the system for actually taking measurements. If we assume that PCOs are the only points in the system where privacy attacks are likely to happen, we can use the measurement system, i.e., a set of selected measurement approaches, to monitor and control the information potentially vulnerable to such attacks. An adversary launches its attacks at a PA through the attack channel into the system, the measurement system measures the privacy values at a PCO through the secure channel. Measurement approach through PCOs is illustrated in Figure 4.1.

We can imagine that the measurements from a system without any privacy protections will have very low privacy values, whereas the measurements form a system with privacy protections should have high privacy values. For our purpose, the measurements from these PCOs provide evidence to verify the architecture developed in PRECIOUSA.

4.2. PCOs in Information Flow Analysis

In D7, we outline the process of information flow analysis in details. We shortly summarize the concept and the required steps to provide background information on how PCOs are determined for the information flow of a given application.
In a first step, the system architecture is modeled in the PRECIOISA layered reference model introduced in D7. The layered meta model consists of a number of views that serve different purposes: the entity view gives a general overview on the participating entities and their relations; the deployment view adds detail by describing the available devices that can be used to deploy functionality on; the logical function view depicts the functional units of the system; and the information flow view models the flow of privacy relevant information through the system, and is consequently the focus of information flow analysis.

Then, the application is analyzed for potential privacy attacks in accordance with a specific adversary model. Attack trees are used as an attack analysis methodology.

Based on the modeled views and the identified attacks, points of control and observation (PCOs) can be identified for the information flow of an application. Thereby, PCOs are placed at spots in the system which are suitable to measure and verify the privacy of the modeled application design. To achieve this, privacy-relevant information items are identified in the information flow with the attack analysis results, and privacy goals are defined for them. The privacy goals and the potential privacy attacks are then matched to the information flow. A PCO is set at each point in the information flow where privacy goals may collide with potential attacks. Additional PCOs are required when information leaves the entity it originated from, e.g. the vehicle, or at domain borders. Most importantly, PCOs have to be placed at positions in the system where information is being modified, processed, stored, or combined with other information, to measure that sufficient privacy is provided.

Once all required PCOs have been identified, they can be evaluated and the privacy level of the given application design can be measured. Section 4.3 will elaborate further on how to apply the privacy metrics discussed in Chapter 3 to PCOs and how to determine the overall privacy value of a design.

The four outlined steps can be repeated for alternative system designs. Either to compare different alternatives in terms of privacy friendliness, or to iteratively improve the privacy level of a certain application design.
4.3. Measurement Based on PCOs

Having identified a set of PCOs in the system, different metrics can be applied at the PCOs to measure their respective privacy level. This measurement can either take place when designing the system or during runtime or both. The choice about when evaluation takes place depends on the specific application in question. It is influenced by whether enough information or solid assumptions can be made during design time to apply a given metric. The following will give some recommendations on when different metrics can be applied.

Basically, Chapter 3 has identified two classes of possible metrics which are either based on \( k \)-anonymity or entropy measurements. The former is more simple to apply, however results can be misleading because of the implicit assumption that each individual in an anonymity set is equally likely to be the origin of a given message in the set. In a first step, entropy based measurement techniques can be used to give additional information about the notion of being anonymous in a \( k \)-anonymity set. Going further, entropy measurement can be used combining the observations of several PCOs to express more complex attacks on privacy.

4.3.1. \( k \)-anonymity Measurement at PCOs

One advantage of \( k \)-anonymity based measurement is that it can be applied to a single PCO always providing almost immediate result that quantifies privacy as an integer. The downside is however that this integer can change over time. Section 3.2.2 already detailed an algorithm for calculating the size of an anonymity set at a PCO, given tuples containing time and location. If applied at design time, the algorithm needs as input the typical – or even better the minimum – number of vehicles passing a defined location interval in a defined time interval. The size of the interval is governed by the granularity of the application-specific time and location information which is sent. Another influence on the size of the anonymity set can be the type of identifier – if any – that passes at a given PCO. If pseudonyms are used and frequently changed, even exact location and time information does not necessarily lead to an anonymity set of size 1. In this case the information observed at a PCO can still be reasonably indistinguishable. This automatically implies that messages containing exact time, location and identity information will lead to an anonymity set of size 1 until messages are modified in some privacy enhancing way during the information flow.

First and foremost, the quality of applying this metric will then rely on the quality of the assumptions on the number of vehicles passing certain areas in a given time. In city scenarios and maybe even highways, good assumptions may be possible but in rural areas there might be no useful assumptions possible. If assumptions are possible, they will be directly applicable during design time because all other input parameters to the \( k \)-anonymity measurement are governed by application specifications.
4.3.2. Entropy Measurement at PCOs

The result of the $k$-anonymity measurement is an anonymity set size which implies that each entity contained in the anonymity set is equally likely to be the origin of a message. This situation, however, is the best case scenario from a privacy viewpoint. If an attacker has further knowledge in addition to observing all messages belonging to an anonymity set then the anonymity of each individual in the set is reduced, although the size of the set is static. The quantification of this notion has been introduced in Section 3.3 using entropy. To calculate the entropy value corresponding to an anonymity set at a given PCO, a model of potential additional attacker knowledge is needed. This can e.g. necessitate that the system is observed for a longer period of time at a given PCO or that more than one PCO are combined for measurement.

One model for attacker knowledge leading to an entropy based privacy measurement is described in Section 3.3.2. Here, a system is modelled by a set of individuals, a set of origins, and a set of destinations. All these vertices sets are linked by directed weighted edges giving the probability that an individual began a trip at a certain origin, that an origin can be tracked to a destination, or that a destination can be mapped to an individual respectively. Without any knowledge, each combination is equally likely. Figure 4.2 shows how the data observed at a PCO during a certain time frame can now serve to model different probabilities representing the attacker knowledge. A user requesting routing information can e.g. lead to successful linkage of an origin-destination pair thus giving the corresponding edge a probability of 1. In addition to that, a priori knowledge of an attacker may have to be assumed which cannot be observed at a given PCO. This could for example be assumptions about how likely different individuals arrive at certain destinations or start at certain origins.

4.4. Measurement of System Privacy

In Chapter 3, we have discussed the system-wide applicability of $k$-anonymity and entropy-based measurement approaches. The identified PCOs are distributed all over the cITS. This means we can eventually select a set of PCOs in the system to measure the same
information, using the same measurement approach. The result reveals the privacy value of the whole system.

However, it is reasonable to assume that information might be changed when it flows in the system. A part of the information might be removed at one point, additional information might be added at another point. Moreover, information might be anonymized or perturbed when privacy-protection mechanisms are implemented in the system. As a result, the measurements from PCOs at different parts of cITS might have different values.

To correctly measure the privacy level of the whole system, several approaches might be possible:

- Calculate the average value, or the mean of measurements from different PCOs.
- Use the minimum value as the system privacy value, since privacy is only as good as the lowest value in a composed system.
- Present the privacy value of a composed system as a continuous space between the maximum and minimum measurements from the PCOs.

Each of these approaches has its pros and cons. It will depend on the actual process of architecture verification and privacy monitoring in the next step of the project to decide which approach best suits the needs. For example, these approaches and potential others can be employed to the actual architecture and the results can be analyzed to determine which measurement reflects system privacy the best.
5. Complementary Aspects to Privacy

When providing a certain degree of privacy protection, most privacy mechanisms inevitably incur extra overhead or reduce the quality of the data under protection. The impacts of privacy protection mechanisms on the system and data usability should also be considered. The measurements of such impacts caused by the privacy protection mechanisms are discussed here.

5.1. Cost of Anonymization

In Chapter 3 we showed how to measure privacy in a given data set. Before sending or releasing values in the data set we aggregated them, for example, to reach a higher level of anonymity. This generalization can be achieved by defining a generalization function \( g_T : T \rightarrow T^* \) which assigns the given data set \( T \) to another data set \( T^* \). There can be different functions \( g_T \) which produce the same measure of privacy, e.g. the same \( k \) for \( k \)-anonymity. Thus, the objective is to choose the best function, i.e. the function with the lowest cost \( c(g_T) \). Theoretical studies denote this as the \( k \)-anonymity problem [51].

Given a data table \( T^* \) and value generalization hierarchies for all attributes in \( T^* \). An example of a hierarchy was presented in Figure 3.3 on page 32. The cost of a generalization function \( g_T \) can be computed as the summed up costs of all values in the tuples of \( T^* \). The cost of generalizing the value of attribute \( A_j \) to level \( h \) in the value hierarchy of attribute \( A_j \) is \( h/H_{A_j} \), where \( H_{A_j} \) is the height of the hierarchy. Let \( t^*[A_j] \) be the projection of tuple \( t^* \) on the attribute \( A_j \) and \( h(t^*[A_j]) \) be the height of the value \( t^*[A_j] \) in the hierarchy of attribute \( A_j \), then the total cost is the sum over all tuples \( t^* \) and attributes \( A_j \) of the release table.

\[
c(g_T) = \sum_{t^* \in T^*} \sum_{j=1}^{m} \frac{h(t^*[A_j])}{H_{A_j}}.
\]

The objective of this definition is to hide the minimum amount of information in the release. Or in other words, reduce information detail only as much as required.

**Example** Let Table 3.4 on page 33 be the data set then all values are generalized to the first level in the corresponding hierarchy. Since the hierarchy of attribute Zipcode has 3 levels and the hierarchy of Age has 2 levels the cost of generalizing each tuple is \( \frac{1}{3} + \frac{1}{2} = \frac{5}{6} \). Note that the bottom level of each hierarchy is level 0 which means that there is no generalization. The cost of generalizing the hole table \( T \) is \( 7 \cdot \frac{5}{6} = \frac{35}{6} \).
Aggarwal et al. [51] showed that the $k$-anonymity problem is NP-hard. On the positive side, they provided an $O(k)$-approximation algorithm for the problem and gave improved positive results of a 1.5-approximation algorithm for the special case of 2-anonymity, and a 2-approximation algorithm for 3-anonymity. Park and Shim [52] showed an $O(\log k)$-approximation.

5.2. Utility Measurements

In the privacy environment, personal data is often communicated or released in aggregated form, i.e. data values are not revealed directly but generalized to less specific and semantically consistent values. For example, instead of sending the exact position of a vehicle in a message this local information is mapped to a quadrant containing this vehicle. The objective is to achieve a predefined degree of anonymity because every vehicle in this quadrant sends the same location information. Obviously, if the quadrants are enlarged more vehicles sends the same location information and the privacy level increases. The disadvantage is that the information becomes more inexact and therefore less useful for the recipient. Imagine, a service provider wants to predict traffic jams, the information that there are lots of vehicles in a particular street is much more useful than that there is high traffic somewhere in the center of town. Thus, the privacy level directly affects the utility of information. In the following we give several interpretations and measurements of data utility.

We explain the techniques based on the running example presented in Tables 3.3 and 3.4 on page 33, and Figure 3.3. Recall, Table 3.4 presents the released data and Figure 3.3 the used value generalization hierarchy.

**Generalization height** A first and easy approach to measure utility was introduced by Samarati [53] and measures for each anonymized value the height of this value in the generalization hierarchies. Intuitively, it is the total number of generalization steps that were performed. The intuition behind this idea is that a generalization step represents a loss of information and the goal is to use as few generalization steps as possible. In Table 3.4 all values are generalized to the first level of the corresponding hierarchy, so each attribute Zipcode and Age is generalized once. A problem with this approach is that not all generalization steps are created equally. A generalization step on one attribute may put many more tuples into an anonymized group than a generalization step on another attribute [32].

**Average size of QI-groups and Discernibility** The next two metrics take into account the sizes of the anonymized groups. The simplest understanding is to measure the average size of the QI-groups generated by the anonymization algorithm [32]. Another metric was introduced by Bayardo and Agrawal [54] and is called discernibility. The idea is to assign a cost to each tuple based on how many other tuples are indistinguishable from it. In general the cost of a tuple is the number of tuples in its anonymized QI-group. Thus,
discernibility is the sum of the squares of the QI-groups, which is $3^2 + 2^2 + 2^2 = 17$ for Table 3.4.

**General Loss Metric (LM)** In order to measure information loss Iyengar [55] introduced a metric called general loss metric (LM). Information in all the potentially identifying columns are assumed to be equally important. So the total information loss due to generalizations and suppressions can be computed by summing up a normalized information loss for each of these columns. The information loss for a column can be computed as the average loss for each entry in the column. The LM can be applied for categorical and numerical values. Given a generalized table $T^*$ and a value generalization hierarchy VGH, let $M_v$ be the number of leaf nodes in the subtree rooted by $v^*$ in VGH and $M$ be the total number of leaf nodes in VGH. Then, the LM for one categorical values $v^*$ in $T^*$ is

$$LM(v^*) = \frac{M_v - 1}{M - 1}.$$ 

In the numerical case a value $i$ is generalized to an interval $i^*$. Let $U_i$ and $L_i$ the upper and lower bound of the interval $i^*$ and $U$ and $L$ the upper and lower bound of the interval of all values in VGH, respectively. Then, the LM for one generalized value $i^*$ in $T^*$ is

$$LM(i^*) = \frac{U_i - L_i}{U - L}.$$ 

We compute the LM for the last tuple of Table 3.4: (1042*, 30–39, Münsterplatz). Of course, attributes Zipcode and Age are numerical but at first we treat them as categorical. For the Zipcode value $v^* = 1042^*$ the LM is $\frac{2-1}{6-1} = 0.2$ because there are two leaf nodes in the subtree rooted by $v^*$ in Figure 3.3 (10421 and 10422) and six leaves at all. Similar, the LM for the Age value $v^* = 30–39$ is $\frac{3-1}{6-1} = 0.4$. If we consider the attributes to be numerical the LM of interval $i^* = 1042^*$ is $\frac{39-30}{10429-10420} = \frac{9}{103}$ and for $i^* = [30 – 39]$ the LM is $\frac{39-30}{39-20} = \frac{9}{19}$. We can observe that the LM is always between 0 and 1.

**Classification Metric (CM)** The next metric also introduced by Iyengar [55] is applicable when somebody wants to train a classifier over the anonymized data and is therefore assigned to the data mining domain. It considers a possible use for the data and is called classification metric (CM). The approach is to treat one attribute as a class label and assign classes $\text{class}(t^*)$ to each tuple $t^*$. Then, a penalty value of each tuple $t^*$ is computed. If the tuple is suppressed the penalty is set to 1. If the majority class label in its QI-group differs from its own class label the penalty is also set to 1. In all other cases the penalty is 0. The CM of the anonymized table $T^*$ is defined as

$$CM = \frac{\sum_{t^* \in T^*} \text{penalty}(t^*)}{|T^*|}.$$
Consider the microdata in Table 5.1 and its 3-anonymous version presented in Table 5.2. Imagine, attribute Sex is the classification attribute then the class labels for the first, second, and sixth tuple are Male and for other tuples gets label Female. The majority class label of the first QI-group of Table 5.2 is Male because there are two tuples with value Male and only one with value Female. Thus, the penalty of the first and second tuple is set to 0 and of the third tuple to 1. For a similar reason the penalty of the last tuple is 1.

The classification metric of the hole table is $CM = \frac{0+0+1+0+0+1}{6} = \frac{1}{3}$.

**Information Loss** The last utility measure we present in this document is called information loss and is an extension of the general loss metric. There are several definitions of information loss we present the definition of Wong et al. [56]. They consider two aspects the information loss caused by the anonymization and the importance of attributes. Let $|t^*|_A$ be the number of values that can be generalized to $t^*|A|$ (the value of attribute $A$ in tuple $t^*$) and $w_A$ be an user defined weight of information loss of attribute $A$. Then the information loss of one tuple $t^*$ is defined as

$$IL(t^*) = \sum_{A \in QI} \frac{|t^*|_A - 1}{|\text{domain of } A| - 1} \cdot w_A$$

and the information loss of the hole table $T^*$ is

$$Dist(T, T^*) = \frac{\sum_{t^* \in T^*} IL(t^*)}{|T^*|}.$$ 

If we use a value generalization hierarchy and set $M_v = |t^*|_A$ and $M = |\text{domain of } A|$ the only differences between $IL$ and $LM$ are the weights. Consider one tuple $t^*$ of the first QI-group of Table 3.4 with $QI = \{\text{Zipcode, Age}\}$, then $t^*|\text{Zipcode}| = 1012*$ and $|t^*|\text{Zipcode}| = 3$ because there are three values in Figure 3.3 which can be generalized to value 1012*. Similar $|t^*|\text{Age}| = 3$. Assume we choose the weights $w_{\text{Zipcode}} = 0.4$ and $w_{\text{Age}} = 0.6$ then the information loss of $t^*$ is $IL(t^*) = \frac{2}{3} \cdot 0.4 + \frac{2}{3} \cdot 0.6 = 0.4$. Since there are 7 tuples in Table 3.4 the information loss of the whole table is $\frac{1}{7} \cdot ((\frac{2}{5} \cdot 0.4 + \frac{2}{5} \cdot 0.6) \cdot 3 + (\frac{6}{5} \cdot 0.4 + \frac{2}{5} \cdot 0.6) \cdot 2 + (\frac{4}{5} \cdot 0.4 + \frac{2}{5} \cdot 0.6) \cdot 2) = 0.331$.
**Discussion**  In this section we presented six metrics to measure utility of a data set. The first three measurements (generalization height, average size of QI-groups, and discernibility) are easy to adopt but have several shortcomings. The main disadvantage is that none of them takes the information loss into account caused by the generalization of values. The general loss metric (LM) eliminates this by considering how many values can be generalized to a given value. User specified importance of attributes are added in the classification metric (CM) and information loss (IL). Since the CM is only applicable for one classifying attribute the information loss is a more flexible way to measure the utility of an anonymization.
6. Defining Metrics for System Development Time and Runtime

In this chapter we address the challenge of how to apply privacy metrics defined in system development time in system runtime. Therefore, we introduce a new method for modelling data in a system. Our measurement presented in this chapter is system oriented and can be applied in the communication and storage domain. A detailed discussion of this model is presented in D6.

Our approach for measuring the system privacy is to model the information needed to infer a certain privacy violating information. The intuition of \( k \)-anonymity is that the combination of some attributes \( A_1, \ldots, A_n \) can uniquely identify some individuals. Therefore, the values of these quasi-identifiers will generalized in the following. We can model this relationship with a functional dependency \( A_1, \ldots, A_n \rightarrow ID \) where \( ID \) is an identifying attribute. Furthermore, there are probably other attributes \( A_{i,1}, \ldots, A_{i,m} \) which determine a quasi-identifier \( A_i \). The \( ID \) attribute itself dominates the sensitive attribute \( S: ID \rightarrow S \). Figure 6.1 presents a graphical overview.

![Figure 6.1.: Model](image)

<table>
<thead>
<tr>
<th>Zipcode</th>
<th>Age</th>
<th>Sex</th>
<th>Current Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>10121</td>
<td>21</td>
<td>M</td>
<td>Wall Street</td>
</tr>
</tbody>
</table>

Table 6.1.: One tuple without ID attribute

<table>
<thead>
<tr>
<th>Name</th>
<th>Zipcode</th>
<th>Age</th>
<th>Sex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andy</td>
<td>10121</td>
<td>21</td>
<td>M</td>
</tr>
</tbody>
</table>

Table 6.2.: The quasi-identifier for Andy

With the help of this representation we are able to measure the amount of data needed to violate privacy. For example in Table 6.1 there are three quasi-identifier attributes Zipcode, Age, and Sex. If an adversary knows all these three attribute values 10121, 21, and M, he is able to conclude that this tuple belongs to Andy (with the help of Table 6.2). For a given system we can observe which attributes are stored in which subcomponent. If attribute Zipcode is stored by service provider \( P_1 \) and attributes Age and Sex are stored by service provider \( P_2 \), both of them may not exchange non-anonymized data because...
then the sensitive values can be linked to a specific individual (we assume that there is always a sensitive attribute which can easily be linked to a subset of quasi-identifiers).

In order to measure privacy we measure the attribute values needed to be used as a quasi-identifier and therefore to be linked to an individual. In Figure 6.1 there are three quasi-identifier attributes $A_1, A_2, A_3$ which build one quasi-identifier $QI := \{A_1, A_2, A_3\}$. If the values of $A_1$ are stored by service provider $P_1$, an adversary needs two more values to perform a linking attack. If attribute $A_2$ is also stored in another service provider $P_2$ the privacy measure drops. If all attributes are present in the system a privacy violation is possible.

**Measurement approach** We measure the potential risk of a privacy violation by counting the attributes that build a quasi-identifier and observing their occurrences in the system. The measure of privacy is the number of attributes which are not present in the system divided by the total number of attributes (the size of the quasi-identifier).

\[
\text{System Privacy} = \frac{\text{number of attributes not in system}}{|QI|}
\]

We introduce weights for the attributes in order to reflect the difficulty of getting the value of the attribute. For example the weights of those attributes for which we can easily obtain values from several sources are set to low values whereas special and not common attributes get higher weights.

\[
\text{System Privacy} = \frac{\sum_{i} w_{A_i}}{\sum_{i=1}^{n} w_{A_i}}
\]

This metric can be applied in design and runtime development. Consider Figure 6.1 and let attributes $A_1$ and $A_2$ be potentially stored in a system and let the weights of the attributes be $w_{A_1} = 0.3$, $w_{A_2} = 0.2$, and $w_{A_3} = 0.5$. Then the system privacy at development time is

\[
\text{System Privacy}_{\text{development}} = w_{A_3} = 0.5
\]

Further if we assume that the value of attribute $A_1$ of individual $i_1$ and the value of attribute $A_2$ of individual $i_2$ is currently stored in the system, we can define those individuals privacy during runtime. The system privacy of $i_1$ and $i_2$ in runtime is given by

\[
\text{System Privacy}_{\text{runtime}}(i_1) = w_{A_2} + w_{A_3} = 0.7 \text{ and } \text{System Privacy}_{\text{runtime}}(i_2) = w_{A_1} + w_{A_3} = 0.8.
\]

The privacy measure of the hole system is of course the minimal value for the individuals. In our example the measure is 0.7. This gives an easy metric to measure an individual’s privacy if the data stored in the system is known.
7. Conclusion

The main object of this deliverable is to develop privacy measurement approaches for cooperative ITS. The measurement approaches provide us a valuable tool box to evaluate and verify the privacy-enforceable architecture and the effectiveness of privacy-protection mechanisms developed in PRECIOUSA. Besides, the measurement approaches also provide inputs to the capability of privacy monitoring of the system in the runtime environment. Our work and achievement to develop suitable measurement approaches for cITS are described in this deliverable.

To measure privacy, we first review the related measurement theory and identify the object to be measured, i.e. measuring privacy as anonymity or unlinkability. We then identify and collect requirements on privacy measurements and common metrics in general IT systems. To take possible differences on how different groups participating in IT systems define privacy into consideration, requirements and corresponding metrics from a wide range of perspectives are discussed. Based on this assessment, two general privacy measurement approaches for cITS are identified: $k$-anonymity and entropy-based measurement. Both approaches are equally applicable to the communication domain and the data storage domain. This gives us the possibility to measure the privacy of the system as a whole by applying different metrics for each system component. We then discuss how to apply the measurement approaches to actual cITS through points of control and observation (PCOs).

However, privacy measurement approaches are only part of the work. We also define several complementary aspects to privacy and discuss how to measure them. Finally we define a first metric to measure both an individual’s privacy and overall system privacy using a $k$-anonymity based approach.

In the next step, the measurement approaches in this deliverable will serve as a basis for the development of verification support tools.
8. Bibliography


A. Oracle security solutions

As a market leader in the data storage domain, these security solutions are representative of the current industrial standards.

**Oracle Database Vault**  Oracle Database Vault provides a transparent solution for mitigating the risk of insider threats and complying with regulatory requirements. Oracle Database Vault uses a number of technical real time access controls:

- Realms – Prevent highly privileged users from accessing application data
- Multi-Factor Authorization– Create trusted paths to data, defining who, when, where and how applications, data and databases are accessed
- Command Rules – Enforce operational policies based on IT Security and internal or external auditor recommendations
- Separation of Duty – Control administrative actions within the database to prevent actions that may violate regulations and best practices
- Reports – Run security related reports on attempted realm violations and other Database Vault enforcement controls

**Oracle Advanced Security**  Oracle Advanced Security provides transparent, standards-based security that protects data through:

- data-at-rest encryption – Oracle Advanced Security Transparent Data Encryption (TDE) uses industry standard encryption algorithms and built-in key management to provide transparent encryption of sensitive application data. TDE automatically encrypts data before it is written to disk and automatically decrypts data before it is returned to the application. The encryption and decryption process is completely transparent to applications and users. TDE offers the ability to protect at the individual attribute level or at the full table level.
- Network encryption – Oracle Advanced Security protects privacy and confidentiality of data over the network. All communication with an Oracle Database can be encrypted with Oracle Advanced Security.
- Strong authentication services – Oracle Advanced Security provides strong authentication solutions leveraging a business’s existing security framework including Kerberos, Public Key Cryptography, and RADIUS.
**Oracle Audit Vault**  
Oracle Audit Vault automates the consolidation of audit data into a secure repository, enabling efficient monitoring and reporting. Built on Oracle's industry-leading technology, Oracle Audit Vault uses Oracle data security to protect audit data end-to-end. Moreover, it provides powerful built-in reports to monitor a wide range of activity including privileged user activity and changes to database structures. Oracle Audit Vault provides security personnel with the ability to detect and alert on activities that may indicate attempts to gain unauthorized access and/or abuse system privileges. Oracle Audit Vault continuously monitors the audit data collected, evaluating the activities against defined alert conditions.

**Oracle Configuration Management**

- Get instantaneous views of IT assets, configuration items, topologies and their dependencies, including: Operating System Configuration, Hardware Configurations, Database Configurations, Application Server Configurations, Packaged Application Configurations.

- Analytics And Reporting Track. Analyze, report and view details related to IT configurations.

- Change Detection. Manage configuration drift through ad hoc, one to one or one to many configuration compare. Schedule comparisons with ‘gold configurations’ or saved baselines.

- Compliance Assessments. Utilize the Compliance Dashboard and Policy groups to receive an at-a-glance view of how specified systems are complying with best practices, security, storage or use defined policies.

- Real-Time Configuration Change Detection. Detect changes to IT configurations in real-time, providing the ability to collect, detect and report change data from a broad range of systems, databases, applications and other IT infrastructure components.

- Facilitate Compliance With Security And Governance Policies, measure and track IT organization's ability to successfully meet its compliance goals. Enable IT departments to easily demonstrate compliance to government regulations and industry standards.

**Oracle Total Recall**  
Oracle Total Recall provides easy to use, transparent retention of historical data.

- Easy to configure – You can easily enable historical data capture for one table or all tables.

- Efficient performance and storage – The capture process is efficient minimizing performance overhead.

- Complete protection from accidental or malicious update – No one, not even administrators, can update historical data directly.
• Archived table data can seamlessly be viewed at any prior point in time.

**Oracle Data Masking** The Data Masking Pack can help organizations comply with privacy and confidentiality laws by masking sensitive or confidential data in development, test or staging environments. The Data Masking Pack uses an irreversible process to replace sensitive data with realistic-looking but scrubbed data based on masking rules and ensures that the original data cannot be retrieved, recovered nor restored. The Data Masking Pack helps maintain the integrity of the application while masking data.

**Oracle Label Security** Oracle Label Security helps organizations address security and compliance requirements using sensitivity labels such as confidential and sensitive. Sensitivity labels can be assigned to users in the form of label authorizations and associated with operations and objects inside the database using data labels. Label authorizations provide tremendous flexibility in making access control decisions and enforcing separation of duty. Oracle Label Security can be used to address numerous operational issues related to security, compliance and privacy.

**Oracle Secure Backup** Oracle Secure Backup, centralized tape backup management solution, provides performing data protection for heterogeneous file systems and the Oracle database. Highly scalable, Oracle Secure Backup’s client-server architecture offers single-point management of distributed UNIX, Linux, Windows, Network Attached Storage (NAS) and tape devices. Safeguarding data and access to the backup domain, Oracle Secure Backup has embedded proven Secure Socket Layer (SSL) technology achieving two-way server authentication and in-transit encryption critical in networked environments.

Bringing enterprise to the Cloud, the Oracle Secure Backup Cloud module extends database backup from local disk and/or tape to the Cloud (Amazon S3 storage).

**Summary** Industry provides set of security solutions meeting all common security and privacy requirements. Using these solutions, it is possible to build privacy compliant data storage for all existing and future privacy regulations. Therefore, from an industrial perspective, the level of data security depends on how the ‘general principles’ are implemented in terms of the state-of-the-art security solutions in the industry.