Objective privacy : understanding the privacy impact of information exchange

In de Eilandje 1 dissertatie-serie verschenen:

Sebastiaan de Hoogh, Design of Large Scale Applications of Secure Multiparty Computation: Secure Linear Programming
Meilof Veeningen, Objective Privacy: Understanding the Privacy Impact of Information Exchange

Nog te verschijnen dissertaties: Peter van Liesdonk, Dion Boesten

---

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-3623-8

This research is supported by the research program Sentinels (`www.sentinels.nl`) as project 'Identity Management on Mobile Devices' (10522). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

Cover: Andrzej Wróblewski, *Striving Towards Excellence*, 1952, the collection of Van Abbemuseum, Eindhoven, courtesy of the Andrzej Wróblewski Foundation

---

Objective Privacy

Understanding the Privacy Impact of Information Exchange

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven,
op gezag van de rector magnificus prof.dr.ir. C.J. van Duijn, voor een commissie
aangewezen door het College voor Promoties, in het openbaar te verdedigen
op dinsdag 10 juni 2014 om 16:00 uur

door

Meilof Geert Veeningen

geboren te Utrecht

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. E.H.L. Aarts
1e promotor:	prof.dr. S. Etalle
copromotor(en):	dr. B.M.M. de Weger
	dr. N. Zannone
leden:	dr.ir. L.A.M. Schoenmakers
	dr. C. Palamidessi (INRIA - Campus de l'Ecole Polytechnique)
	prof.dr. P. Samarati (Università degli Studi di Milano)
	prof.dr. W.H. Hesselink (Rijksuniversiteit Groningen)

Contents

1
Introduction

Contents

IMAGINE A GROUP of eight Dutch hospitals that need an electronic system for distributing patient data to researchers. Because patient data are privacy-sensitive, medical researchers should work on anonymised patient data. However, data about the same patient should be collected from different hospitals, and it should be deanonymisable in case the researcher finds out something that is relevant for the patient. One proposal for this system involves hospitals pseudonymising the data using a cryptographic hash function (intuitively, a function that is easy to compute but hard to invert) before sending it to a "central infrastructure" that distributes it, repseudonymised, to researchers. Another proposal is to use a "pseudonymisation service" that performs pseudonymisation using a cryptographic construction based on a well-protected secret. From the point of view of patient privacy, which proposal would you pick?

1.1 Information Exchange in Distributed Systems

In the above example, privacy-sensitive information is exchanged in a *distributed system*. In general, a distributed system is a software system in which components located on networked computers communicate and coordinate their actions by passing messages[1]. Often, this network is the Internet, and the components are operated by different organisations (in this case, the hospitals, the central infrastructure, the pseudonymisation service and the researchers). In addition to the above example, distributed systems that exchange possibly privacy-sensitive information include identity management

[1] Coulouris et al. (2005)

systems[2]. In such systems, one party (the *service provider*) receives identity information endorsed by another party (the *identity provider*) to whom a user has authenticated. Other examples are electronic voting systems in which voters register at an administrator, and cast their votes at a counter; or road toll pricing systems, in which cars communicate their location to "toll service providers", which aggregate results so that toll chargers can send bills.

[2] Hansen et al. (2004)

Message passing in a distributed system is done using *communication protocols*. Such protocols specify what information should be exchanged in what order and format. Typically, such protocols use (combinations of) cryptographic techniques for various objectives, e.g., to ensure that messages in transit are not tampered with or read by third parties. For instance, in the example of patient data pseudonymisation, transmitting the cryptographic hash of a patient identifier instead of the identifier is meant to prevent the patient identifier from being leaked[3]. Many different cryptographic techniques exist[4], and they usually need to be combined (e.g., an encryption of a hashed message) for the objectives of the protocol to be achieved, often in elaborate and subtle ways. For instance, suppose a message is signed by some party and then encrypted: then the recipient knows that the party signed the message but not that he encrypted it, hence the message could originally have come from a different protocol. However, if the message is first encrypted and then signed, the recipient knows that the party signed the encryption, but not that he knew the original message, hence the signer may have inadvertently signed the wrong message.[5] Which choice is appropriate depends on the goals that the system needs to achieve. Hence, the design of communication protocols is both crucial for achieving the goals of the distributed system, and non-trivial to understand.

[3] In the system proposed in Parelsnoer Initiatief (2008), see Chapter 8

[4] E.g., encryption, cryptographic hashes, digital signatures: see Menezes et al. (1996); but also more complex techniques like authenticated key agreement, anonymous credentials, and zero-knowledge proofs

[5] Davis (2001)

1.2 *Privacy Impact of Information Exchange*

As more and more personal information is exchanged in distributed systems, privacy risks are becoming more and more of a concern. There have been numerous reports of information from such systems being used for secondary purposes, or being stolen and abused by third parties. Legislation (e.g., EU Directive 95/46/EC, HIPAA) attempts to reduce these risks by requiring such systems to satisfy the *data minimisation* principle. That is, systems have to be designed to ensure that actors in such systems collect and store only the minimal amount of personal information needed to fulfil their task. This includes making sure that actors only learn identity attributes that they actually need (*data secrecy*). It also includes making sure that actors in the system cannot identify the data subject if there is no need for them to do so (*anonymity*); or even, that they cannot tell if different transactions involve the same data subject if they do not need to know (*unlinkability*). In addition, data minimisation not just involves preventing single actors from gaining such knowledge; it also means preventing *coalitions* of different actors from being able

to correlate their separate knowledge. Note that these concerns all relate to knowledge of legitimate actors in the system rather than outside attackers; in fact, a recent report on computer crime shows that 44% of all reported security incidents are due to such insider abuse[6].

[6] Richardson (2008)

However, whether a system respects these data minimisation concerns, depends crucially on how information is exchanged using communication protocols. For instance, consider an identity management scenario where a service provider receives identity information endorsed by two different identity providers. Depending on the design of the system, these identity providers may or may not learn which service provider obtains the identity information; and the service provider may learn some but not all identity attributes about a user. Also, suppose that the user wants to remain anonymous, so she does not provide any identifying information (e.g., address, phone number) to the service provider. Depending on the design of the system, the service provider may or may not be able to identify her by teaming up with one of the identity providers and checking their communication logs for shared identifiers, e.g., session identifiers. In many areas, *privacy-enhancing* communication protocols[7] have been designed that specifically aim to guarantee data minimisation. Namely, such protocols use cryptographic primitives to ensure that participants learn as little information as possible, and that they have as little ability as possible to correlate information from different sources. Privacy-enhancing protocols have been proposed for a wide range of applications: e.g., smart metering, e-voting, and electronic toll collection.

[7] See Troncoso (2011) for a good overview

1.3 Understanding Privacy Impact of Information Exchange

Understanding the privacy differences between different protocols for information exchange is important, e.g., for system designers who want to use privacy-enhancing protocols, or for system architects who want to select what protocols to use. However, existing approaches are not sufficient for obtaining this understanding, as we argue below.

High-level comparisons miss interesting privacy differences. Existing comparisons of privacy impact in different systems are often performed in a high-level and informal way. For instance, the Independent Centre for Privacy Protection Schleswig-Holstein[8] presents a large-scale comparison of identity management systems, in which one privacy criterion is the "usage of pseudonyms/anonymity"; it is judged on a "yes/no" scale. This general criterion fails to take into account questions like whether the same pseudonym is shared between different identity providers, or between the identity and service provider. Another criterion is that the "user [is] only asked for needed data": this does not take into account, for instance, which parties see the data on the way from the identity provider to the ser-

[8] Independent Centre for Privacy Protection Schleswig-Holstein (2003)

vice provider, or whether the system only allows the disclosure of full attributes ("age") or also of properties of these attributes (">18"). Each of these unconsidered questions reveals interesting privacy differences between proposed privacy-enhancing identity management systems[9]. Moreover, high-level comparisons like the one above are typically performed informally based on high-level system architectures, rather than rigorously based on the actual communication that takes place. Although this is sufficient for performing a high-level assessment, it is not sufficient for performing a comparison that takes into account the above unconsidered questions, and that does so in a precise and verifiable way.

[9] In particular, the identity management systems by Bangerter et al. (2004), Chadwick and Inman (2009), Vossaert et al. (2011): see Chapter 7

Privacy analysis at the level of cryptographic primitives is difficult. Unfortunately, it is not straightforward to perform a more precise and verifiable analysis of privacy issues. The main reason for this is that protocols typically use combinations of cryptographic primitives such as encryption and digital signatures in elaborate ways. Hence, an understanding of the privacy impact of information exchange starts with an understanding of the cryptography underlying the communication protocols used.

Fundamentally, many cryptographic primitives used in communication protocols are designed and analysed using the concept of *provable security*[10] in the computational model. Intuitively, properties of these cryptographic primitives are proven by showing that, if a certain adverse situation occurs (e.g., somebody without the decryption key can decrypt an encrypted message), then this violates some well-defined assumption (e.g., no computer can factor large numbers into their prime factors in reasonable time). Privacy-like properties can be captured with an "ideal" functionality[11] that describes what all protocol participants should learn; primitives can be rigorously proven to "implement" this ideal functionality (in the presence of any attacker with some well-defined capabilities), which implies in particular that they do not learn any additional information. Although these techniques were designed to analyse isolated primitives, some theory has been developed to reason about communication protocols in which multiple primitives are combined[12]. However, these techniques are very technical, low-level, and hard to automate; and moreover, they only cover cryptographic primitives designed especially with the techniques in mind. Unfortunately, this does not cover very many primitives in use today. Hence, these techniques are not yet sufficiently practical or general to analyse privacy in existing systems.

[10] One of the seminal works in this direction is Bellare (1998)

[11] E.g., Beaver (1991)

[12] The nowadays standard framework for such analysis is from Canetti (2001)

Formal methods require encoding privacy properties. Formal methods approaches have been proposed to analyse various properties of communication protocols. Such approaches check for logical errors in the use of cryptographic primitives, rather than errors in the design (as above) or implementation of these primitives themselves. Cryptographic primitives are modelled as "black boxes" with

a simplified, abstract functionality[13]. Messages containing cryptographic primitives (e.g., encryption, digital signature) are described as abstract "terms", and an explicit enumeration is provided of the operations that actors can perform on them (e.g., decryption, signature verification). By modelling communication protocols in this way, various security properties can be expressed, and, in many cases, automatically verified[14]. For instance, this includes "secrecy" properties[15] stating that, whatever operations an attacker performs on the messages he knows, he cannot learn some particular secret that the protocol aims to hide.

[13] The seminal paper in this field is Dolev and Yao (1981); much current research is based on the applied pi calculus: see Abadi and Fournet (2001), Blanchet et al. (2008)

[14] Available verification tools include AVISPA (Armando et al. (2005)), ProVerif (Blanchet and Smyth (2011)), and Tamarin (Schmidt et al. (2012))

[15] Two influential works on defining them are Abadi (1998), Blanchet (2004)

To use formal methods techniques for evaluating the privacy impact of information exchange, we need to encode privacy properties of distributed systems as properties of sets of terms representing cryptographic messages. This is not trivial. For instance, suppose we want to encode whether an identity provider and service provider have a common session identifier they can use to link their knowledge about some user. We cannot simply check secrecy of the session identifier (as above) in their separate sets of known messages, because messages known by one actor may help to derive information from messages known by the other. We also cannot simply check secrecy of the session identifier in their combined set of known messages, because the actors can only link the identifier if they know that it occurs in *both* sets of messages. Intuitively, when using formal methods, we need to encode privacy properties by capturing that a particular piece of information can be derived from a particular message.

Existing encodings are not general enough, and hard to verify. Nowadays, the standard way of performing this encoding is by means of equivalences[16]. The idea is to consider two sets of messages which coincide except on privacy-sensitive information. For instance, to consider if an identity provider and service provider can combine their knowledge about a user using a shared identifier, we consider a service provider who is involved in two transactions. In the first set of messages, the first transaction uses the shared identifier and the second one does not; in the second set of messages, the second transaction uses the shared identifier and the first one does not. If the actors can "see the difference" between the two sets of messages (formally, the two sets are not "statically equivalent"[17]), then we conclude that they can use the shared identifier to combine their knowledge. These equivalences are quantified over arbitrary attacker behaviour by modelling interacting actors as "processes", typically using the applied pi calculus[18].

[16] Some important works in this direction are Blanchet et al. (2008), Delaune et al. (2009), Arapinis et al. (2010), Dong et al. (2013)

[17] Abadi and Fournet (2001)

[18] Abadi and Fournet (2001)

Although many privacy properties have been verified with this approach, there are two reasons why it is insufficient for understanding privacy impact of information exchange. The first reason is that, so far, the encoding by means of equivalences is performed on an ad-hoc basis depending on the particular protocol. For instance, in the above example, the definition of the privacy property depends

on which message components are identifiers. This is a problem because this makes it impossible to compare systems by defining properties independently from a system, and then verifying them by automatically encoding them as equivalences. Some works have partially addressed this problem by defining general encodings. Arapinis et al.[19] propose general definitions for linking identifiers from different protocols, but only consider the identifier of the sender of a message, rather than identifiers of the data subject whom communicated information is about. Dong et al.[20] propose general definitions for privacy of a particular piece of information, but do not consider linking information. Fundamentally, an encoding powerful enough to capture all privacy aspects would seem to require "annotating" information with whom it is about, and whether or not it is an identifier, something existing approaches do not do.

[19] Arapinis et al. (2010)

[20] Dong et al. (2013)

The second, more practical reason is that encodings of privacy properties as equivalences are hard to verify. Existing encodings are typically defined in terms of *observational equivalence*[21], for which ProVerif[22] is the main available verification tool. Although observational equivalence is a very powerful property (in particular, it considers attackers, which is beyond the scope of this thesis), it is also too complex for automated verification. To still prove observational equivalence in some cases, ProVerif applies a rather blunt over-approximation that fails to cover many processes that are actually observationally equivalent. Even with this over-approximation, in many cases it does not terminate. As a consequence, for some simple protocols, a more or less comprehensive sets of privacy properties can be verified[23], but for more complicated protocols, only the analysis of knowledge of particular actors is possible[24]. In any case, the need to formalise equivalences carefully to ensure termination makes it hard to combine this approach with an automated encoding. Hence, although many useful results have been obtained using the standard encoding approach using equivalences, this approach is not sufficiently general or automatable to perform comprehensive privacy analysis.

[21] Blanchet et al. (2008)

[22] Blanchet and Smyth (2011)

[23] E.g., Arapinis et al. (2012)

[24] E.g., Dong et al. (2012)

1.4 Research Question

Motivated by the gap between, on the one hand, high-level and informal privacy comparisons between various systems, and, on the other hand, precise but incomplete and incomparable results for particular systems, we aim to answer the following research question in this thesis:

How can we rigorously understand the privacy impact of information exchange in distributed systems?

The aim of this thesis is to develop techniques for obtaining such an understanding. To answer the research question, we need techniques that satisfy three basic requirements. To make our analysis

rigorous, the techniques need to provide precise and verifiable results (requirement 1). To make our analysis useful, these results need to be easy to interpret (requirement 2). On the other hand, to make analysis feasible in practice[25], it should be largely automated (requirement 3).

[25] In particular, because we need to verify properties about multiple actors and coalitions of actors

In this thesis, we aim to contribute to answering the research question by presenting a set of techniques based on ideas from the formal methods approaches discussed above. We divide the question into three sub-questions that we subsequently aim to answer:

Question 1. How can we represent privacy properties about actors in distributed systems in a system-independent way?

As discussed above, to compare distributed systems designed for the same purpose, we need to be able to represent privacy properties as properties of messages in a way that does not depend on the particular system. This representation should be precise (requirement 1) and easy to interpret (requirement 2). The above research question addresses this need.

Question 2. How can we automatically decide privacy properties based on a formal model of information exchange?

Given a privacy property, we then need to decide whether it holds given a formal model of messages. The above question asks for such a decision procedure. By basing it on a formal model, we satisfy the verifiability part of the first requirement. By asking for an automated procedure, we address the third requirement.

Question 3. Which steps need to be followed to actually analyse privacy impact of information exchange?

While the first two questions are theoretical, a full answer to our research question should also discuss the more practical aspects of actually performing a privacy impact analysis using our techniques. Starting from a set of systems for information exchange in a particular application domain (e.g., identity management), it should be clear what steps need to be taken to perform such an analysis, and how these steps work in practice. This third question covers this concern.

With the above research plan, we focus on the choice of communication protocol, i.e., we compare the extent to which different protocols satisfy the data minimisation principle. In particular, we do not consider privacy impact that is due to the semantics of the information exchanged, because this cannot be influenced by the protocols — e.g., we do not consider how combinations of attributes like address, city of birth, and age might be used to identify people. Also, we consider only threats by insiders, i.e., legitimate actors in the system; as argued, privacy breaches by insiders are indeed a major concern. As a consequence, we do not consider attackers who try to break into the system.

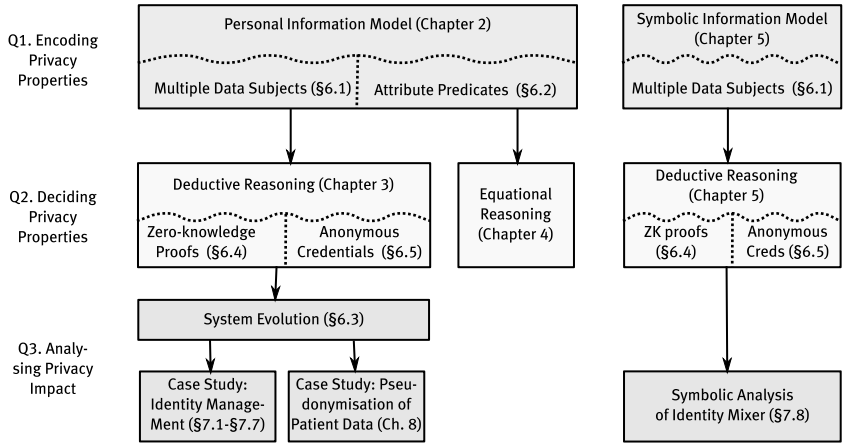


Figure 1.1: Systematic overview of the contributions of this thesis

1.5 Contributions

To answer the above questions, we make the following contributions, systematically shown in Figure 1.1 along with references to the relevant chapters.

To answer Question 1, we propose the *Personal Information Model*: a model of knowledge about personal information that allows for system-independent specification of privacy properties. We define a basic model that is sufficient for many applications, and show how privacy properties can be specified as properties of this model. We also extend it with the *multiple data subjects* extension to model pieces of information with multiple data subjects, and the *attribute predicates* extension to model boolean predicates that attributes may satisfy. Although the Personal Information Model is not dependent on the system, it is dependent on characteristics of the scenario (e.g., the number of parties involved and the amount of personal information exchanged). We present an alternative model, the *Symbolic Information Model*, that generalises the previous model to make it scenario-independent. Also for this model, the Multiple Data Subjects extension is defined. Hence, the Personal Information Model and the Symbolic Information Model allow system-independent encoding of privacy properties. This provides our answer to Question 1.

To answer Question 2, we provide three alternative mechanisms by which privacy properties can be automatically decided. First, we propose an approach to populate the Personal Information Model (and hence, to verify privacy properties defined in the model) based on *deductive reasoning*. This approach relies on formal models of cryptographic primitives: we present models from the literature for common primitives; but we also propose our own models for *zero-knowledge proofs* and *anonymous credentials* for use with the deductive reasoning approach. The deductive reasoning approach is limited in what kind of primitives can be accurately modelled; therefore, we propose an alternative approach to populate the Personal Informa-

tion Model based on *equational reasoning*. With this approach, many more models of primitives from the literature can be used. Finally, we show how the *deductive reasoning* approach can be used to populate not just the Personal Information Model, but also the Symbolic Information Model. (As a consequence, our models of *zero-knowledge proofs* and *anonymous credentials* can also be used in the symbolic setting.) For the two deductive reasoning approaches, we propose algorithms and implementations for automated privacy verification. For the equational approach, we show how properties can be decided with the help of existing tools.

In summary, given a formal model of communication, and a set of privacy properties specified using the Personal Information Model, we give two automated ways of deciding whether they hold: namely using deductive and equational reasoning. We also present an automated way to decide privacy properties in the Symbolic Information Model. This provides our answer to Question 2.

Finally, **to answer Question 3**, we show the steps needed to perform a privacy analysis using two concrete case studies. We first show how to obtain a formal model of messages from a model of communicating actors by proposing the *system evolution* formalism. We then present case studies in the domains of *identity management* and *pseudonymisation of patient data*. These two case studies are of independent interest. For identity management, we contribute a new and comprehensive set of privacy requirements; and new formal models of four different identity management systems. For patient data pseudonymisation, we contribute a rigorous analysis of achievable privacy guarantees. The two case studies demonstrate two ways in which our techniques can be used to perform privacy analysis: by verifying a given set of properties, and by visually comparing privacy in different systems. Both case studies are performed using the (scenario-dependent) Personal Information Model: we also present an analysis of one identity management system, *Identity Mixer*, that uses the (scenario-independent) Symbolic Information Model. We present the case studies in a systematic way, so that the presented steps also apply to other privacy analyses. This is our answer to Question 3.

1.6 Reading Guide

Given the overlap and interdependency between the contributions listed above, we think it wise to provide some suggestions on how to navigate this thesis. To this end, we present several possible "tracks" depending on the reader's interest (Figure 1.2).

Our first two tracks give the reader a full overview of our analysis framework from theory to practice; they represent the two ways in which a privacy analysis using our framework can be done. The **Visual Comparison track** demonstrates how our framework can be used to visually compare privacy, in the setting of pseudonymising patient data for research purposes. After the introduc-

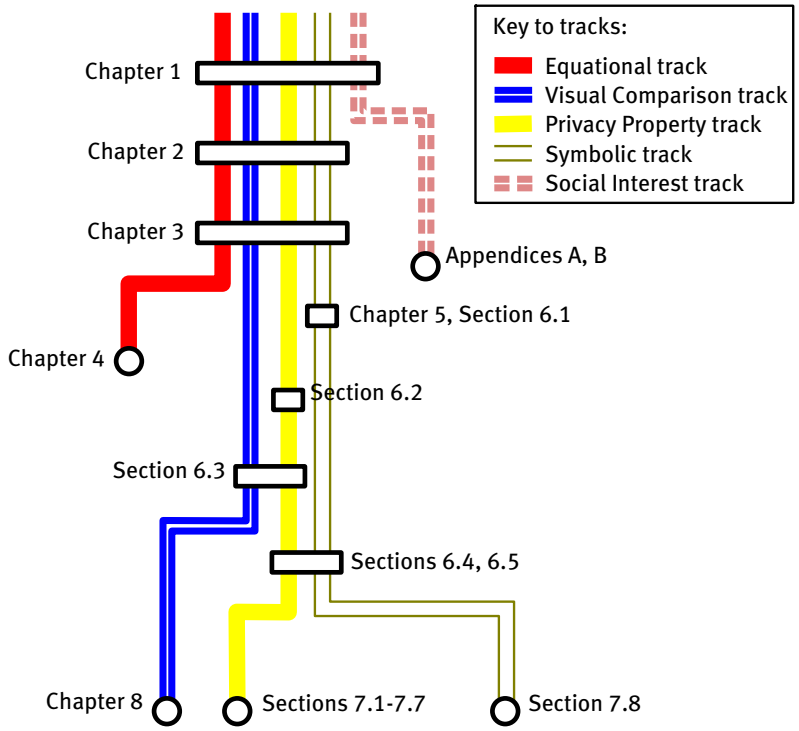


Figure 1.2: Reading guide for this thesis (not including related work and conclusions)

tion, this track goes through Chapters 2 and 3 describing the Personal Information Model and deductive reasoning. The track then briefly passes through Section 6.3 on system evolution, before arriving at Chapter 8, in which the pseudonymisation case study is discussed. The **Privacy Property track** shows how our framework can be used to formulate privacy properties once, and then verify them for multiple systems, in an identity management case study. As the visual comparison track, this track goes trough Chapters 2 and 3 on the Personal Information Model and deductive reasoning, and through Section 6.3 on system evolution. However, it also passes through three extensions needed to model and analyse the case study: Section 6.2 on attribute predicates; Section 6.4 on zero-knowledge proofs; and Section 6.5 on anonymous credentials. Finally, this track arrives at Sections 7.1–7.7, where the case study is described.

For people with a more theoretical inclination, we suggest the **Symbolic track**. After passing Chapters 2 and 3 on the Personal Information Model and deductive reasoning, this track visits Chapter 5, in which we generalise the Personal Information to the Symbolic Information Model, and show a formal link between the two models. This track then continues towards an application: after visiting some needed extensions (Sections 6.1, 6.4, and 6.5)[26], it arrives in Section 7.8, which discusses an analysis of the Identity Mixer identity management system using the Symbolic Information Model.

For people who know about, or are interested in, modelling cryptographic primitives using equational theories, we suggest the **Equational track**. After going through Chapters 2 and 3, this track directly terminates in Chapter 4, in which we propose an alternative

[26] And, perhaps, briefly exploring Sections 7.1–7.7

to our deductive reasoning model using equational theories; and in which we formally establish a link between the two alternatives.

Finally, for people who are no more than superficially interested in the topic of this thesis: you have already made it to the end of Chapter 1! We now suggest you follow the **Social Interest track** directly to Appendices A and B, in which I very briefly (and hopefully, relatively accessibly) summarise the remainder of this thesis, and provide a nice overview of the trips I made while working on its material.

Eindhoven, April 2014

2
Personal Information Model

Contents

WHEN PERSONAL INFORMATION is exchanged in a communication system, each actor in the system typically has a different partial view on that information. For instance, consider the scenario Figure 2.1, in which Alice sends a message to Bob via Eve, containing the passport number and birth date of Steve. To protect this message, Alice has encrypted it using some key k that she has shared with Bob beforehand. Hence, both Alice and Bob know the contents of this message. Eve, who has passed on the message, does not have key k, so she does not learn the contents of the message; however, if she shares it with malicious Mallory who has somehow obtained key k, they can together learn the passport number and birth date, and maybe even link it to Steve's photo which Mallory had already stolen before.

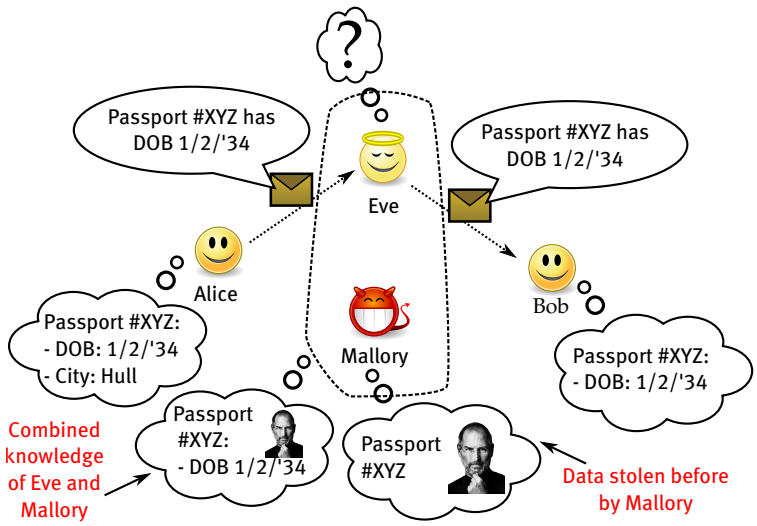


Figure 2.1: A simple communication system, in which different actors have different partial views on the personal information exchanged

Using the formalisms developed in this chapter, we can precisely express which actors and coalitions of actors in the above example hold which personal information. The goal of this thesis is to present tools for the analysis of knowledge about personal information. The formalisms in this chapter provide a precise and comprehensive representation of this knowledge. This representation is used later to verify if particular information systems satisfy particular "privacy properties", and to compare the privacy of different systems to each other. Hence, we design the representation to be expressive enough to capture all interesting aspects of an actor's knowledge, but also to be amenable to automatic computation. Basically, the formalism is a list of all "pieces of information" that the actor knows, grouped according to his knowledge of which of these pieces of information are about the same person. To define privacy properties, it will be convenient to refer to information in terms of where it was obtained (e.g., "the identifier of the user in protocol instance X should be unknown"), so our notation will capture this. In addition, to verify privacy properties, it will be relevant to know the contents of pieces of information (e.g., attributes of a different type may nonetheless have the same contents), so we will also capture that.

In modelling personal information and knowledge about it, we will make two main assumptions:

- Discrete information — There is a finite set of pieces of personal information that each belong to a particular data subject. Each piece of information has a well-defined contents. (However, different pieces of information may have the same contents.)

- Discrete knowledge — Actors may or may not be able to learn these pieces of information; and they may or may not be able to learn that these pieces of information are about the same data subject. In both cases, we do not allow uncertainty: either an actor knows a piece of information or a link, or he does not.

The above abstractions are common in the protocol verification literature[1], and simplify both the specification of properties and the modelling of protocols. At the end of this chapter, we discuss approaches that do not make these abstractions.

[1] E.g., see Meadows (2003) for a survey

Outline In this chapter:

- We introduce the *Personal Information (PI) model* (§2.1): a formalism that describes personal information in an information system at a certain point in time;

- We introduce the *view* on this PI Model of an actor involved in the system (§2.2) that captures the knowledge about this information held by that actor;

- We show how various *privacy properties* (§2.3) can be modelled as properties of items from these views;

- We present a visualisation called *coalition graphs* (§2.4), in which the knowledge about personal information of all actors and coalitions of actors in the system are summarised;
- We discuss limitations and possible extensions of our model (§2.5).

2.1 Personal Information Model: Information in the System

The Personal Information (PI) Model is a formalism to model all personal information in an information system at a certain point in time.

Personal Information

A piece of personal information in the PI Model represents a *specific* value that has a *specific* meaning as personal information about a *specific* person. For instance, it can represent "the age of Alice" (with contents "22") or the social security number of Bob (with contents "132-13-0398"). We distinguish between two types of digital personal information: *identifiers* and *data items*. Identifiers are unique within the system (e.g., Bob's social security number); for data items, this is not necessarily the case (e.g., Alice's age). The sets of identifiers and data items are denoted $\mathcal{I}^{\text{inf}}$ and $\mathcal{D}^{\text{inf}}$, respectively.[2] Elements of the set $\mathcal{O}^{\text{inf}} := \mathcal{I}^{\text{inf}} \cup \mathcal{D}^{\text{inf}}$ are called *personal items*. We partition $\mathcal{O}^{\text{inf}}$ according to which personal items are about the same person; the *related* equivalence relation $\Leftrightarrow$ on $\mathcal{O}^{\text{inf}}$ indicates which personal items are in the same equivalence set.

[2] The reason for using $*^{\text{inf}}$ in the notation will become apparent later

However, the above model of personal information is insufficient to model all privacy aspects of communication protocols that we are interested in. First, it is relevant to know whether different pieces of information have the same contents or not. For instance, Alice's age may be the same as Bob's, and Alice's age may be the same as Alice's apartment number. Whether this is the case influences what information can be determined from cryptographic primitives: for instance, an actor can determine a piece of information from its cryptographic hash if he knows another piece of information with the same contents. Second, it is relevant to distinguish between different "representations" of information that an actor learned at different moments. Namely, an actor may learn the same piece of information (e.g., "the age of Alice") twice (e.g., in two protocol instances with different session identifiers) without realising that it is the same information. Possibly, one of these representations can be combined with other privacy sensitive information, but the other representation can not. In this case, only knowledge about the former representation is relevant from a privacy point of view. To analyse such a situation, we need to be able to differentiate between the knowledge of the actor about the former and latter representation of the information.

Three-Layer Model

Above, we modelled pieces of information, and argued that we additionally need to capture different representations of these pieces of information, as well as their contents. Hence, we introduce a three-layer model of personal information. Pieces of information, as defined above, are at the middle *information layer*, e.g. "the city that Alice lives in". The top *context layer* of the model distinguishes between different representations of information by describing the context in which a piece of information has been observed, e.g., "the city of the user in protocol instance #1". The bottom *contents layer* of the model describes the actual value of a pieces of information, e.g., "Eindhoven". Actor knowledge is described using the context layer and reasoned about using the contents layer. The information layer is used to specify privacy properties independently from any particular context-layer representations; and to visualise analysis results (see Section 2.4).

At the context layer, a representation of a piece of information is described in terms of the context in which it has been observed. More precisely, a context-layer representation of a piece of information is a *variable* belonging to a *profile* belonging to a *domain*. A *domain* is any separate digital "place" where personal information is stored or transmitted. For instance, domain η may represent a database and domain π an instance of a communication protocol. A *profile* represents a particular data subject in a domain. For instance, profile 231 in domain η may represent an entry about one person in database η, or profile *cli* in domain π may represent the person performing the logical role "client" in communication protocol π. The combination of a domain π and profile *cli* represents a particular data subject and is called a *context*, denoted, e.g., $*|^{\pi}_{cli}$.[3] Finally, a *variable* represents a particular piece of information about the data subject in the profile. A variable describes the piece of information in terms of the role it has in the profile, e.g. session identifier *id* or age attribute *age*. The combination of a domain π, profile *cli* and variable *id* represents a particular piece of information, and is called a *context personal item*, denoted, e.g., $id|^{\pi}_{cli}$.

[3] Profiles themselves are not unique, e.g., the clients in communication protocols π, π' may both have profile *cli*. Also, different profiles in a domain may represent the same data subject, e.g., duplicate entries in a database.

The set of all context personal items is denoted $\mathsf{O}^{\mathsf{ctx}}$. We distinguish between context-layer representations of identifiers, called *context identifiers* $\mathsf{I}^{\mathsf{ctx}} \subset \mathsf{O}^{\mathsf{ctx}}$, and context-layer representations of data items, called *context data items* $\mathsf{D}^{\mathsf{ctx}} \subset \mathsf{O}^{\mathsf{ctx}}$. Although we focus primarily on information in protocol instances, it is usually insightful to also model information from other sources, e.g., databases. Namely, this way we can analyse whether it is pissible to combine information from the protocol with information from, e.g., the database.

At the contents layer, the contents of pieces of personal information are represented as bitstrings $\in \{0,1\}^*$. In fact, for our purposes the exact representation is not relevant; it suffices to know which pieces of information have the same contents, and which do not.

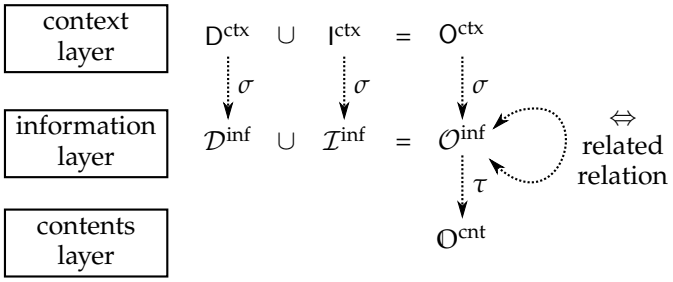


Figure 2.2: Symbols in Definition 2.1.1 and their relations: single-headed arrows denote maps between the different layers; the double-headed arrow represents the $\Leftrightarrow$ relation on $\mathcal{O}^{\mathrm{inf}}$

2

Maps Between Layers and Formal Definition

Apart from these descriptions of pieces of personal information at three layers, the PI Model also defines mappings between the layers. Namely, it defines a mapping σ from the context layer to the information layer; and a mapping τ from the information layer to the contents layer. Properties of σ and τ reflect characteristics of the different pieces of information, as shown below. Formally, a PI Model is defined as follows (see Figure 2.2 for a visual summary of all notation):

Definition 2.1.1. A *Personal Information (PI) Model* is a tuple

$$(\mathsf{O}^{\mathrm{ctx}}, \mathcal{O}^{\mathrm{inf}}, \mathbb{O}^{\mathrm{cnt}}, \Leftrightarrow, \sigma, \tau)$$

such that:

- $\mathsf{O}^{\mathrm{ctx}}$ is a set of *context personal items* of the form $v|_a^\kappa$. Here, v is called the *variable*, κ is called the *domain*, and a is called the *profile*. $\mathsf{O}^{\mathrm{ctx}}$ is partitioned into *context data items* $\mathsf{D}^{\mathrm{ctx}} \subset \mathsf{O}^{\mathrm{ctx}}$ and *context identifiers* $\mathsf{I}^{\mathrm{ctx}} \subset \mathsf{O}^{\mathrm{ctx}}$ (i.e., $\mathsf{O}^{\mathrm{ctx}} = \mathsf{D}^{\mathrm{ctx}} \cup \mathsf{I}^{\mathrm{ctx}}$, $\mathsf{D}^{\mathrm{ctx}} \cap \mathsf{I}^{\mathrm{ctx}} = \varnothing$);
- $\mathcal{O}^{\mathrm{inf}}$ is a set of *personal items*, partitioned into sets $\mathcal{D}^{\mathrm{inf}} \subset \mathcal{O}^{\mathrm{inf}}$ of *data items* and $\mathcal{I}^{\mathrm{inf}} \subset \mathcal{O}^{\mathrm{inf}}$ of *identifiers* (i.e., $\mathcal{O}^{\mathrm{inf}} = \mathcal{D}^{\mathrm{inf}} \cup \mathcal{I}^{\mathrm{inf}}$, $\mathcal{D}^{\mathrm{inf}} \cap \mathcal{I}^{\mathrm{inf}} = \varnothing$);
- $\mathbb{O}^{\mathrm{cnt}} \subset \{0,1\}^*$ is a set of *contents items*;
- $\Leftrightarrow$ is an equivalence relation on $\mathcal{O}^{\mathrm{inf}}$ called the *related* relation;
- σ is a map $\mathsf{O}^{\mathrm{ctx}} \to \mathcal{O}^{\mathrm{inf}}$ such that:
  1. $\sigma(\mathsf{I}^{\mathrm{ctx}}) \subset \mathcal{I}^{\mathrm{inf}}$ and $\sigma(\mathsf{D}^{\mathrm{ctx}}) \subset \mathcal{D}^{\mathrm{inf}}$;
  2. $\sigma(x|_k^\kappa) \Leftrightarrow \sigma(y|_k^\kappa)$ for all $x|_k^\kappa, y|_k^\kappa \in \mathsf{O}^{\mathrm{ctx}}$;
- τ is a map $\mathcal{O}^{\mathrm{inf}} \to \mathbb{O}^{\mathrm{cnt}}$ that is injective on $\mathcal{D}^{\mathrm{inf}}$, i.e., for any identifiers $i, j \in \mathcal{I}^{\mathrm{inf}}$: if $\tau(i) = \tau(j)$, then $i = j$.

The first three bullets define information at the context, information, and contents layers, respectively. The fourth bullet defines personal relations at the information layer. The fifth and sixth bullet define the mapping between the three layers: we demand that σ preserves the type of information and the personal relations implied by contexts; and that τ ensures that the contents of identifiers are unique.

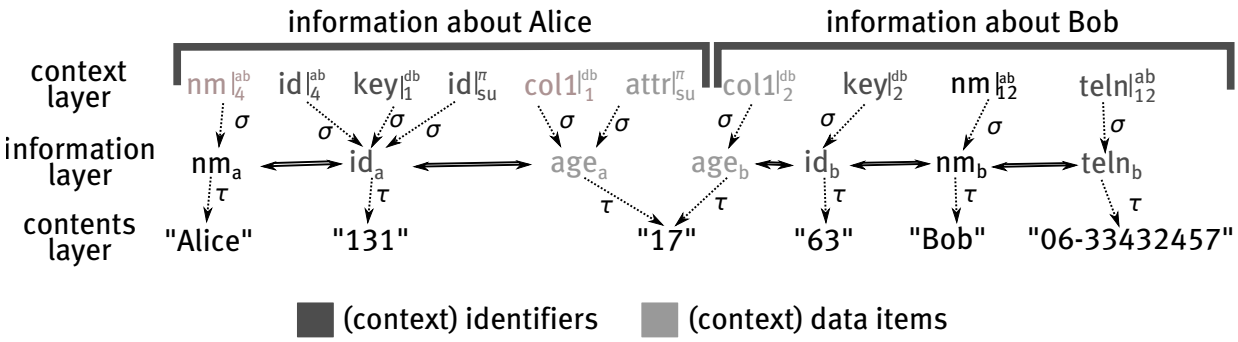


Figure 2.3: Personal Information Model of Example 2.1.2

We introduce notation for context personal items $x|_k^\eta$, $y|_l^\chi$ representing the same contents. Namely, if $\tau(\sigma(x|_k^\eta)) = \tau(\sigma(y|_l^\chi))$, then we write $x|_k^\eta \doteq y|_l^\chi$ and we call them *content equivalent*.

The next example shows a PI Model representing all personal information in a particular scenario.

Example 2.1.2. Figure 2.3 shows a PI Model representing personal information about two persons, Alice and Bob, in a simple scenario. In this scenario, a client and a server exchange information about Alice. Namely, the server has a database with personal information about different persons; the server and client engage in a protocol to exchange information about Alice; and the client combines the results with her address book. The PI Model captures this information as well as the context it occurs in.

At the information layer of this PI Model, Alice has identifier id_a, name nm_a and age age_a; Bob has identifier id_b, name nm_b, age age_b, and telephone number $teln_b$. Alice and Bob happen to be of the same age, so $\tau(age_a) = \tau(age_b)$; the other pieces of information have distinct contents.

At the context layer of this PI Model, the personal information in this scenario is modelled as follows:

- domain db (database held by the server): Each profile $k \in \{1, 2\}$ in this domain represents a database entry consisting of database key $key|_k^{db}$ and column value $col1|_k^{db}$. As shown in the figure, the keys and column values map to the data subjects' identifiers and ages, respectively.
- domain ab (address book of the client): Each profile $k \in \{4, 12\}$ in this domain represents an entry in the address book. The fourth entry of the address book contains name $nm|_4^{ab}$ and identifier $id|_4^{ab}$ (mapping to information about Alice); the 12th entry contains name $nm|_{12}^{ab}$ and telephone number $teln|_{12}^{ab}$ (mapping to information about Bob).
- domain π (protocol instance): The client and server engage in an instance π of a protocol in which identifier $id|_{su}^{\pi}$ and attribute $attr|_{su}^{\pi}$ are exchanged about data subject su; in this case, the subject is Alice and the attribute is her age.

At the contents layer of this PI Model, six different bitstrings

model the contents of the above information. □

2.2 Views: Actor Knowledge

The *view* of an actor captures his partial knowledge about the personal information in a system. In the previous section, we introduced the PI Model to capture all personal information in the system at a certain point in time. The knowledge of an actor at that point in time consists of knowledge of some pieces of personal information from the PI Model, and knowledge that some of these pieces of information are about the same person. Formally, an actor's view consists of a set of context-layer items and an equivalence relation on their contexts:

Definition 2.2.1. Let $M = (\mathsf{O}^{\mathrm{ctx}}, \mathcal{O}^{\mathrm{inf}}, \mathsf{O}^{\mathrm{cnt}}, \Leftrightarrow, \sigma, \tau)$ be a PI Model. A *view* on M is a tuple $V = (\mathsf{O}, \leftrightarrow)$ such that:

- $\mathsf{O} \subset \mathsf{O}^{\mathrm{ctx}}$ is the set of *detectable items* in V;
- $\leftrightarrow$ is an equivalence relation on contexts $*|_k^{\pi}$ of items in $\mathsf{O}^{\mathrm{ctx}}$ called the *associability* relation.

Given two detectable context items $d|_k^{\pi} \in \mathsf{O}$, $e|_l^{\eta} \in \mathsf{O}$, we write $d|_k^{\pi} \leftrightarrow e|_l^{\eta}$, and call the two items *associable*, if $*|_k^{\pi} \leftrightarrow *|_l^{\eta}$.

As argued above, an actor cannot necessarily recognise if two context items $\mathsf{o}_1, \mathsf{o}_2$ represent the same piece of information (in particular, whether or not they are about the same data subject); i.e., if $\mathsf{o}_1, \mathsf{o}_2 \in \mathsf{O}$, then the actor does not necessarily know whether $\sigma(\mathsf{o}_1) = \sigma(\mathsf{o}_2)$. Indeed, his knowledge of whether o_1 and o_2 are about the same data subject, i.e., whether $\mathsf{o}_1 \Leftrightarrow \mathsf{o}_2$, is captured by the associability relation $\leftrightarrow$. By inspecting their contents, the actor *does* know whether $\tau(\sigma(\mathsf{o}_1)) = \tau(\sigma(\mathsf{o}_2))$.

By defining $\leftrightarrow$ on contexts rather than context items, we capture the fact that an actor can always associate context items from the same context. Context items from different contexts can, in our reasoning model (Chapters 3 and 4), be associated by observing that the same identifier occurs in both contexts. Our definition also allows associability between contexts in which no detectable context item exists. This will be useful in defining involvement properties in the next section.

Given a set $\mathcal{A}$ of actors in the information system, we denote the view of actor $a \in \mathcal{A}$ by $V_a = (\mathsf{O}_a, \leftrightarrow_a)$. As mentioned above, the PI Model can contain personal information transmitted in protocol instances as well as any additional information (e.g., databases) held by the actors. Thus, an actor's view on this PI Model captures how he can combine information observed in different contexts. The view of *coalition* $A \subset \mathcal{A}$ is denoted $V_A = (\mathsf{O}_A, \leftrightarrow_A)$. It represents the knowledge of personal information when the actors in the coalition combine all information (e.g., databases, protocol transcripts) they have, and contains at least the knowledge of each individual actor in the coalition.

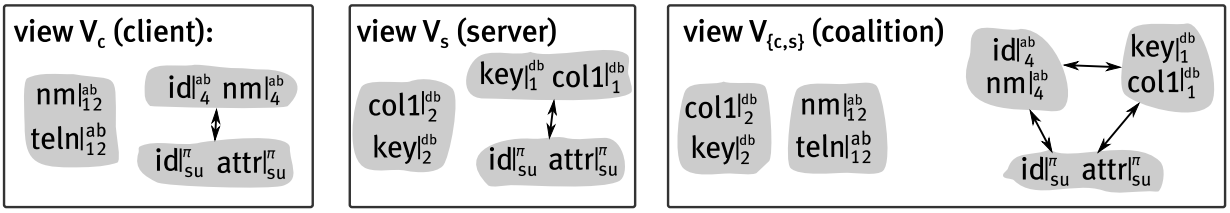


Figure 2.4: Views of actors c and s and coalition $\{c,s\}$ in a scenario (Example 2.2.2). Context personal items shown are detectable; grey areas represent contexts; arrows between grey areas represent the associability relation.

We next show an example of the views that different actors, and coalitions of these actors, can have on a PI Model.

Example 2.2.2. Consider the PI Model M from Example 2.1.2. In the scenario, we are interested in the views of the client and the server on M, as well as the view that the coalition of client and server together may have. These views are denoted $V_c = (\mathsf{O}_c, \leftrightarrow_c)$, $V_s = (\mathsf{O}_s, \leftrightarrow_s)$, and $V_{\{c,s\}} = (\mathsf{O}_{\{c,s\}}, \leftrightarrow_{\{c,s\}})$, respectively. Figure 2.4 shows possible views after some particular communication protocol has been executed (domain π).

First consider the view $V_c = (\mathsf{O}_c, \leftrightarrow_c)$ on M modelling personal information known by the client. This information comprises the entries from her telephone book and the information about Alice that has been communicated. About Bob, the client knows his name $nm|^{ab}_{12} \in \mathsf{O}_c$ and telephone number $teln|^{ab}_{12} \in \mathsf{O}_c$ as an entry $*|^{ab}_{12}$ in her telephone book. In particular, because the two items share the same context, we have $nm|^{ab}_{12} \leftrightarrow_c teln|^{ab}_{12}$, i.e., Alice knows that these who pieces of information are about the same person.

About Alice, the client knows two context-layer representations of identifier id_a: one as part of her telephone book entry ($id|^{ab}_{4} \in \mathsf{O}^c$), and one as a piece of information sent in protocol instance π ($id|^{\pi}_{su} \in \mathsf{O}^c$). She knows the name of Alice as part of the telephone book entry ($nm|^{ab}_{4}$), and she knows the age as transmitted in the protocol ($attr|^{\pi}_{su} \in \mathsf{O}^c$). Moreover, the client can associate the contexts $*|^{ab}_{4}$ and $*|^{\pi}_{su}$; in particular, $nm|^{ab}_{4} \leftrightarrow_c attr|^{\pi}_{su}$, i.e., she knows that the name and age are information about the same person.

The view $V_s = (\mathsf{O}_s, \leftrightarrow_s)$ of the server also contains information about both Alice and Bob. About Bob, the server knows two pieces of information $col1|^{db}_{2}$, $key|^{db}_{2}$ in context $*|^{db}_{2}$ representing a database entry. About Alice, the server similarly knows two pieces of information $col1|^{db}_{1}$, $key|^{db}_{1}$ from the database. In addition, it knows the two other context-layer representations $id|^{\pi}_{su}$, $attr|^{\pi}_{su}$ of that same information as transmitted in the protocol instance π; and it can associate $*|^{\pi}_{su}$ and $*|^{db}_{1}$.

Now consider the view $V_{\{c,s\}}$ of the client and server if they combine their knowledge. In this view, all information about Alice from the two actors is mutually associable, meaning the actor know that it is about the same data subject (in the figure, all contexts representing Alice are connected by arrows). However, information about Bob is divided into two equivalence classes: the client knows name nm_{bob} (as $nm|^{ab}_{12}$) and his telephone number $teln_{bob}$ (as $teln|^{ab}_{12}$) and the server knows age age_b (as $col1|^{db}_{2}$) and identifier id_b (as $key|^{db}_{2}$), but

they cannot associate this information to each other (indicated by the absence of arrows between the information in the figure). □

2.3 *Verifying Privacy Properties using Views*

We intend our model of knowledge to be expressive enough so that relevant privacy properties from the literature can be verified by inspecting actor views. This includes both "functional properties" modelling what *should* be learned by the actors in the protocol, and "privacy properties" modelling what *should not* be learned. We now discuss what kinds of properties can be expressed in our model.

The most basic kinds of properties expressible in our model are (un-)detectability and (un-)linkability properties:

- *Un-)detectability properties* — Can a given actor/coalition of actors detect a given context item?
- (Un-)linkability properties — Can a given actor/coalition of actors associate two given contexts?

Apart from these two "explicit" types of information about a person above, also "implicit" information that a person interacts with a certain entity in the system (e.g., a certain hospital, or a local branch of a bank) may be privacy sensitive, especially when combined[4]. To express such information, we can include pieces of information about these entities in the PI Model. For instance, if context π represents a protocol instance in an e-health setting, then $*|_h^\pi$ may represent the hospital involved in the protocol instance[5]. If, in some view $V = (\mathsf{O}, \leftrightarrow)$, the user $*|_u^\pi$ is associable to a context $*|_{alice}^{db}$ and the hospital $*|_h^\pi$ is associable to a context $*|_{umcg}^{\cdot}$, then this reflects the knowledge that the actors represented by $*|_{alice}^{db}$ and $*|_{umcg}^{\cdot}$ were both "involved" in protocol instance π. This motivates the following, third, type of property:

- *(Non-)involvement properties* — Is there a domain d in which an actor can associate one profile to a given context c_1, and another profile to a given context c_2, i.e., does he know that the actors represented by c_1, c_2 were both involved in domain d?

More complex properties can be defined as arbitrary combinations of these elementary properties and their negations. In our case studies (Chapters 7 and 8), we will show that, in practical settings, this includes many interesting properties. In Chapter 9, we compare these properties to other privacy properties from the literature.

The next example shows different types of properties.

Example 2.3.1. We formulate two properties for the scenario given in Example 2.1.2. Recall that we have views

$$V_c = (\mathsf{O}_c, \leftrightarrow_c), V_s = (\mathsf{O}_s, \leftrightarrow_s), \text{ and } V_{\{c,s\}} = (\mathsf{O}_{\{c,s\}}, \leftrightarrow_{\{c,s\}})$$

of the client, server, and coalition of client and server together, respectively. First, since the goal of the protocol is to exchange information, we can check whether the client has indeed learned the age of

[4] See Pashalidis and Meyer (2006) for an analysis of this issue

[5] When modelling information about entities that do not represent real-world persons, the granularity at which this is done depends on the application at hand. For instance, in an e-health system we may consider all information about the same hospital as linked, whereas in a financial system within the hospital, we may need to distinguish between the accounts and cleaning departments within that hospital.

Alice, and whether she can link it to her telephone book entry. This corresponds to verifying that $attr|^{\pi}_{su} \in O_c$ and $attr|^{\pi}_{su} \leftrightarrow_c id|^{ab}_4$ hold (a detectability property and a linkability property, respectively). Second, since the protocol does not concern Bob, we may want to make sure that the client and server together cannot inadvertently link Bob's telephone number and age due to this protocol instance. This corresponds to verifying that $teln|^{ab}_{12} \leftrightarrow_{\{c,s\}} col1|^{db}_2$ does not hold (an unlinkability property).

Now consider the views in the particular system from Example 2.2.2. In this case, both properties hold. Namely, in view V_c, $attr|^{\pi}_{su} \in V_c$ and $age|^{\pi}_{su} \leftrightarrow_c id|^{ab}_4$ are true (Figure 2.4, left), while in view $V_{\{c,s\}}$, $teln|^{ab}_{12} \leftrightarrow_{\{c,s\}} col1|^{db}_2$ is not true (Figure 2.4, right). □

2.4 Coalition Graphs

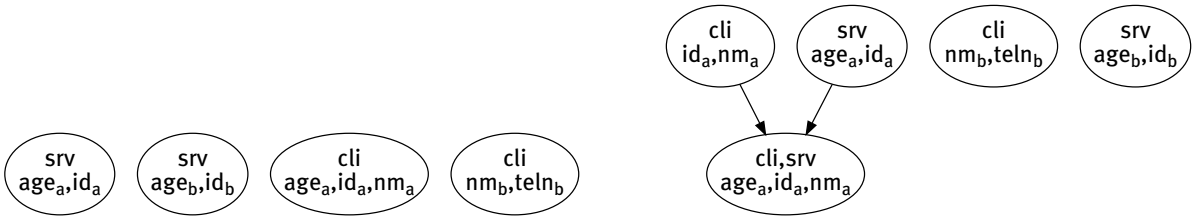
We now propose a visual way of representing the knowledge of all actors in an information system. Recall that, given a PI Model M and a set $\mathcal{A}$ of actors, each *coalition* $A \subset \mathcal{A}$ of actors in $\mathcal{A}$ has a view $V_A = (O_A, \leftrightarrow_A)$ on the personal information in the system. The *coalition graph* of the system visualises these views by showing exactly who can detect and associate what information, while also visualising which actors profit from combining their knowledge with others. To make this visualisation manageable, we represent pieces of information at the information layer rather than considering their representation at the context layer. When inspection of the coalition graph has raised a privacy concern about a particular coalition of actors, the view of these actors (at the context layer) can then be inspected to see exactly how that coalition obtained the personal information.

Intuitively, each node in the coalition graph represents a certain "record" about a person that can be derived by a certain coalition of actors. Namely, suppose we want to visualise knowledge about a set $\mathcal{O}^{ioi} \subset \mathcal{O}^{\text{inf}}$ of personal items, called the *items of interest*. A *record* is a subset $O' \subset \mathcal{O}^{ioi}$. This record is *detectable* by a coalition $A \subset \mathcal{A}$ of actors with view $(\mathsf{O}_A, \leftrightarrow_A)$ if there exists a set of detectable, mutually associable context personal items representing (via σ) the personal items in O'. In this case, we write $A \vDash O'$. We call $A \vDash O'$ *elementary* if there is no smaller coalition $B \subsetneq A$ such that $B \vDash O'$ and there is no larger record $\mathcal{O}^{ioi} \supset O'' \supsetneq O'$ such that $A \vDash O''$. The nodes of a coalition graph are these elementary items $A \vDash O'$; an edge from $A \vDash O'$ to $B \vDash O''$ indicates that, by growing from A to $B \supsetneq A$, coalition A can enlarge its record O' to $O'' \supsetneq O'$:

Definition 2.4.1. The *coalition graph* for set $\mathcal{O}^{ioi}$ of items of interest and collection $\{V_A\}_{A \subset \mathcal{A}}$ of views, is the graph $(W, \leq)$ with:

- $W = \{(A, O') \mid A \subseteq \mathcal{A};\ O' \subseteq \mathcal{O}^{ioi};\ A \vDash O' \text{ holds and is elementary}\}$
- $(A_1, O_1) \leq (A_2, O_2)$ iff $A_1 \subseteq A_2 \wedge O_1 \subseteq O_2$.

When visualising coalition graphs, we label each node (A, O')



(a) After communication

(b) Before communication

Figure 2.5: Coalition graphs for the PI Model of Example 2.1.2: after communication (left; see Example 2.4.2) and before communication (right; see Example 2.4.3)

with one line for coalition A, and another line for record O'. We do not draw self-loops or edges that follow from others by transitivity.

The following two examples show what coalition graphs look like.

Example 2.4.2. Consider the PI Model from Example 2.1.2; set $\mathcal{A} = \{c, s\}$ of actors; and set $O = \{id_a, age_a, nm_a, id_b, age_b, nm_b, teln_b\}$ of items of interest.

In Example 2.2.2, we presented the views $\{V_A\}_{A \subset \mathcal{A}}$ of the client, server, and coalition of both after they have exchanged information about Alice in protocol instance π. The coalition graph corresponding to these views is shown in Figure 2.5(a). As the figure shows, the server can build two records: one containing the age and identifier of Alice ($\{s\} \vDash \{age_a, id_a\}$), and one containing the age and identifier of Bob ($\{s\} \vDash \{age_b, id_b\}$). Similarly, the client can build two records about Alice and Bob, respectively.

In this case, there are no nodes representing records detectable by the coalition $\{c, s\}$. Technically, this is because there are no $O' \subset O$ for which $\{c, s\} \vDash O$ is elementary: each record detectable by the client and server together, is also detectable by one of the actors alone. This reflects that, when the server and client combine their knowledge, they do not discover any new associations between the information they have. Indeed, the client can already detect a record containing all information about Alice in the PI Model; and both the client and the server can detect records about Bob, but they cannot associate them, i.e., $\{c, s\} \nvDash \{nm_b, teln_b, age_b, id_b\}$. □

Example 2.4.3. We again consider the PI Model from Example 2.1.2; set $\mathcal{A} = \{c, s\}$ of actors; and set $O = \{id_a, age_a, nm_a, id_b, age_b, nm_b, teln_b\}$ of items of interest. However, now let us consider the knowledge of these actors *before* they have exchanged information about Alice. Suppose this knowledge is as follows:

$$V_c = (\mathsf{O}_c, \leftrightarrow_c) = (\{nm|_{12}^{ab}, teln|_{12}^{ab}, id|_4^{ab}, nm|_4^{ab}\}, =);$$
$$V_s = (\mathsf{O}_s, \leftrightarrow_s) = (\{col1|_1^{db}, key|_1^{db}, col1|_2^{db}, key|_2^{db}\}, =);$$
$$V_{\{c,s\}} = (\mathsf{O}_{\{c,s\}}, \leftrightarrow_{\{c,s\}}) = (\mathsf{O}_c \cup \mathsf{O}_s, \{*|_4^{ab} \leftrightarrow_{\{c,s\}} *|_1^{db}\}).$$

These views represent that the client just knows the entries from her telephone book, with no associations (i.e., $\leftrightarrow_c$ is equality) and the server just knows the entries from its database, with no associations. Moreover, the client and server together can link their information

about Alice (for instance, by seeing the overlapping identifier) but not about Bob.

The coalition graph corresponding to the above situation is shown in Figure 2.5(b). Here, knowledge about Bob is as in the previous example: both the client and the server have personal information about Bob, but they cannot associate this information if they combine their knowledge. However, about Alice, the situation is different: the client knows id_a and nm_a (node $\{c\} \vDash \{id_a, nm_a\}$) and the server knows age_a and id_a (node $\{s\} \vDash \{age_a, id_a\}$); if they combine their knowledge, they can build a bigger record consisting of all this information: $\{c, s\} \vDash \{age_a, id_a, nm_a\}$. □

We can also use coalition graphs to visually compare the knowledge of actors in different systems, or at different moments in time in the same system. Namely, if $A \vDash O'$ in some system X but not in some system Y, then this suggests that concerning record O', Y offers better privacy. To perform this comparison between X and Y visually, we use a *combined coalition graph* that combines the nodes from the coalition graphs of X and Y; shows for each node $A \vDash O'$ whether O' is detectable in X, in Y, or in both; and keep the same partial relation $\leq$ as before. This idea can be generalised to compare any number of coalition graphs:

Definition 2.4.4. Let $G_{X_1} = (V_1, \leq_1), \ldots, G_{X_n} = (V_n, \leq_n)$ be a finite set of coalition graphs. The *combined coalition graph* $G_{\{X_1,\ldots,X_n\}}$ is the graph $(V, \leq)$ with

$$V = \{(A, O, N) \mid \exists i : (A, O) \in V_i, \\ N = \{i \mid \exists (A', O') \in V_i : A' \subseteq A, O' \supseteq O\}\};$$

and $(A_1, O_1, N_1) \leq (A_2, O_2, N_2)$ iff $A_1 \subseteq A_2 \wedge O_1 \subseteq O_2$.

We visualise combined coalition graphs by labelling each node (A, O, N) with one line specifying coalition A and another line specifying the record O that this coalition can detect; the set N of systems in which detectability holds is visualised by using different styles to draw the nodes. Again, we do not draw self-loops or edges that follow from others by transitivity.

The follow example demonstrates combined coalition graphs.

Example 2.4.5. Consider the coalition graphs G_{before} and G_{after} from Examples 2.4.2 and 2.4.3, respectively. The combined coalition graph of these two graphs is shown in Figure 2.6.

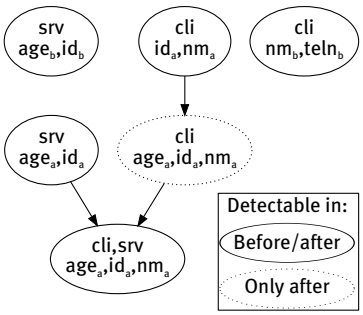


Figure 2.6: Combined coalition graph of the graphs of Figure 2.5

The combined coalition graph contains the nodes of both original coalition graphs; for each coalition A and record O', it indicates whether O' is detectable by A before and/or after communication. For instance, $\{srv\} \vDash \{age_b, id_b\}$ is true both before communication (i.e., in G_{before}) and after (i.e., in G_{after}); $\{cli\} \vDash \{age_a, id_a, nm_a\}$ is true after communication but not before. Note that there are no detections that are true before communication but not after: this makes sense because communication can only increase the knowledge of actors. Note also that detection $\{cli, srv\} \vDash$

$\{age_a, id_a, nm_a\}$ is true both before and after communication, but node $(\{cli, srv\}, \{age_a, id_a, nm_a\})$ does not occur in G_{after}. This is because $\{cli, srv\} \vDash \{age_a, id_a, nm_a\}$ is not an elementary deduction in G_{after}: already the smaller coalition $\{cli\}$ can deduce the same record $\{age_a, id_a, nm_a\}$. □

In Chapter 8, we use combined coalition graphs in a real case study.

2.5 Discussion

The model of knowledge about personal information presented in this chapter can be used to analyse any privacy concern that is expressible in terms of the elementary detectability, linkability, and involvement properties described in Section 2.3. Although, as our case studies (Chapters 7–8) will demonstrate, this includes many relevant properties proposed in the literature, there are also privacy aspects that our model does not capture.

Most significantly, we do not allow reasoning about the meaning of pieces of personal information. For instance, we do not express how different pieces of information about the same person relate to each other, e.g. "address" is a combination of "street name" and "house number". In Chapter 6, we will partially overcome this restriction by modelling that one piece of information represents a "predicate" on another piece of information, which helps in the verification of certain privacy properties. However, other than distinguishing between identifiers and non-identifiers and allowing these predicates, we do not interpret pieces of information in any way. Also, we do not consider (probabilistic) links due to combinations of non-identifying attributes, e.g., matching name and zip code in two different contexts imply a link with high probability. These choices reflect the goal of our approach, namely to compare the privacy effects from using a particular system to exchange personal information (independently from what information exactly is exchanged). On the other hand, to obtain a full understanding of the privacy of users that does take such inferences into account, our approach can be complemented with orthogonal (e.g., probabilistic) methods. We discuss this topic further in Chapter 9.

Our model assumes that pieces of information can have only one data subject. However, certain pieces of information (e.g., a shared key between two parties, or the result of a chess match), do not satisfy this assumption. We generalise our model to allow multiple data subjects in Chapter 6.

Apart from explicitly transferred information, i.e., the user's attributes, we analyse one particular kind of implicitly transferred information; namely, involvement properties. However, other kinds may be of interest as well. For instance, the number of transactions performed by a user may be privacy sensitive, as may be the mere date and time of certain activities (e.g., in the context of smart meter-

ing, as analysed by Rial and Danezis[6]). Knowledge about numbers of transactions can be expressed in our model; date and time may be appended as "tags" to communication.

The model we present here has appeared in several forms in earlier papers by the author[7]. In Veeningen et al. (2014), the model was presented combined with the attribute predicates extension from Chapter 6. In all earlier works, our model included a set $\mathcal{E}$ of "entities" to whom personal information belongs. In this thesis, we implicitly model entities via the "related" relation, mainly for reasons of notational elegance. Coalition graphs were presented before in Veeningen et al. (2012).

[6] Rial and Danezis (2011)

[7] Veeningen et al. (2011a); Veeningen et al. (2011b); and Veeningen et al. (2014)

3
Detectability and Linkability with Deductions

Contents

HAVING INTRODUCED PI MODELS AND VIEWS, we are able to formally model the knowledge of Alice, Eve, Bob and Mallory, whom we met at the beginning of the previous chapter. In this chapter, we formalise the reasoning about messages that determines this knowledge. In particular, we formalise reasoning steps like "[Eve] does not have key k, [so] she does not learn the contents of the message", and "[Eve and Mallory] can together learn the passport number and birth date, and [...] link it to Steve's photo".

Our approach relies on existing deductive techniques for reasoning about cryptographic messages, which we adapt to the setting where these messages contain personal information. Deductive systems are a traditional technique to reason about the knowledge of attackers in the *Dolev-Yao model*[1]. The Dolev-Yao model captures different actors who exchange messages over channels that are under control of an attacker. These messages are modelled as abstract terms (e.g. representing an encryption), and the operations that the attacker can perform on these messages (e.g., decryption using the key) are explicitly enumerated, using a deductive system or similar formalisms. For instance, we can model secrecy of a key k in the Dolev-Yao model by requiring that however the attacker manipulates the messages sent over channels between honest actors, he can never get k by performing operations on the messages he has seen.

[1] Named after the seminal paper by Dolev and Yao (1981)

In the previous chapter, we have defined the knowledge of actors in terms of knowledge not about pieces of information, but about specific representations of these pieces of information (captured

by particular context personal items). Because traditional message reasoning methods do not consider representations, we cannot directly apply them to determine actor knowledge. Another problem with existing methods is that they do not consider the possibility that different pieces of information have the same contents. For instance, a traditional decryption rule could be that "if encryption of message m under key k is known, and key k is known, then message m is known". This rule is defined regardless of whether other pieces of information with the same contents as k are available[2]. In this chapter, we will adapt such rules to let them act on messages that contain personal information as defined in the previous chapter. For instance, the above rule becomes "if encryption of context item $d|_k^\pi$ under key $key|_{srv}^\pi$ is known, and a key with the same contents as $key|_{srv}^\pi$ is known, then $d|_k^\pi$ is known". Carefully translating existing rules, as described above, is already almost enough to reason about personal information in messages. However, to obtain reasonable definitions for detectability and associability, we turn out to additionally need some reasoning on the knowledge that certain messages have the same contents, even when those contents are not known.

In this chapter, we introduce our deduction-based model of personal information in cryptographic messages, and our procedure to determine the views of actors based on the messages they have seen.

[2] For traditional applications such as authentication protocols, messages usually just consist of identifiers and cryptographic material like keys and nonces. In such cases, it is reasonable to assume that no two different pieces of information have the same contents, hence the traditional decryption rule is sufficient. However, when reasoning about privacy-enhancing protocols in which user attributes are transmitted as in this thesis, this assumption is no longer valid.

Outline In this chapter:

- We generalise the PI Model from the previous chapter to an *Information Model*, that additionally includes non-personal information occurring in cryptographic messages such as nonces and keys (§3.1);
- We introduce a formal model of cryptographic messages built from personal and non-personal information (§3.2);
- We present our reasoning model of what information can be derived from these messages (§3.3);
- We show how standard cryptographic primitives like encryption and digital signatures can be captured in this model (§3.4);
- We define detectability and linkability using the above reasoning model, hence obtaining the actor view following from a set of known cryptographic messages (§3.5);
- We present an alternative reasoning system that, under non-restrictive conditions, gives the same result as the model in Section 3.3, while being more suitable for implementation (§3.6);
- We present an algorithm to compute actor views from known messages, based on the alternative reasoning system (§3.7);
- Finally, we discuss validity and expressiveness of our model (§3.8).

3.1 Three-Layer Model of Non-Personal Information

Communication in privacy-enhancing protocols uses messages built up from personal and other information, e.g., private/public keys and nonces. In the previous chapter, we defined the PI Model (Definition 2.1.1) to capture all personal information in the system. In Section 2.3, we showed that the PI Model can also be used to model information (e.g. private/public keys) about non-personal entities (e.g., a hospital or a service provider). Doing this allows for reasoning about the involvement of these actors in protocols, both for the verification of involvement properties (§2.3) and for obtaining symbolic privacy guarantees (Chapter 5).

However, this still leaves information in messages that does not refer to any entity in particular, e.g., session keys and nonces. In order to accurately model messages containing such information, we now extend our model with such non-personal information. As with personal information, we model it using our three-layer model. Although knowledge about different context-layer representations of this information is not directly relevant to privacy, doing this will be advantageous both when defining equatability later in this chapter, and when developing the symbolic variant of our model in Chapter 5.

Formally, we model non-personal information at the context layer by a set $\mathsf{G}^{\mathrm{ctx}}$ of *context non-personal items*. Items in $\mathsf{G}^{\mathrm{ctx}}$ belong to a domain, but not to a profile: in this case we denote the profile as "$\cdot$", e.g. $shakey|_{\cdot}^{\pi}$. At the information layer, we define set $\mathcal{G}^{\mathrm{inf}}$ of *non-personal items*. We obtain a straightforward generalisation of the definition of PI Model (cf. Definition 2.1.1):

Definition 3.1.1. An *Information Model* is a tuple

$$(\mathsf{P}^{\mathrm{ctx}}, \mathcal{P}^{\mathrm{inf}}, \mathbb{P}^{\mathrm{cnt}}, \Leftrightarrow, \sigma, \tau)$$

such that:

- $\mathsf{P}^{\mathrm{ctx}}$ is a set of *context items* of the form $v|_a^{\kappa}$. Here, v is called the *variable*, κ is called the *domain*, and a is called the *profile*. $\mathsf{P}^{\mathrm{ctx}}$ is partitioned into *context non-personal items* $\mathsf{G}^{\mathrm{ctx}} \subset \mathsf{P}^{\mathrm{ctx}}$ (with $a = \cdot$), *context data items* $\mathsf{D}^{\mathrm{ctx}} \subset \mathsf{P}^{\mathrm{ctx}}$ (with $a \neq \cdot$), and *context identifiers* $\mathsf{I}^{\mathrm{ctx}} \subset \mathsf{P}^{\mathrm{ctx}}$ (also with $a \neq \cdot$).
- $\mathcal{P}^{\mathrm{inf}}$ is a set of *information items*, partitioned into sets $\mathcal{G}^{\mathrm{inf}} \subset \mathcal{P}^{\mathrm{inf}}$ of *non-personal items*, $\mathcal{D}^{\mathrm{inf}} \subset \mathcal{P}^{\mathrm{inf}}$ of *data items*, and $\mathcal{I}^{\mathrm{inf}} \subset \mathcal{P}^{\mathrm{inf}}$ of *identifiers*.
- $\mathbb{P}^{\mathrm{cnt}} \subset \{0,1\}^*$ is a set of *contents items*;
- $\Leftrightarrow$ is an equivalence relation on $\mathcal{O}^{\mathrm{inf}}$ called the *related* relation;
- σ is a map $\mathsf{P}^{\mathrm{ctx}} \to \mathcal{P}^{\mathrm{inf}}$ such that 1) $\sigma(\mathsf{G}^{\mathrm{ctx}}) \subset \mathcal{G}^{\mathrm{inf}}$, $\sigma(\mathsf{D}^{\mathrm{ctx}}) \subset \mathcal{D}^{\mathrm{inf}}$, and $\sigma(\mathsf{I}^{\mathrm{ctx}}) \subset \mathcal{I}^{\mathrm{inf}}$; 2) $\sigma(x|_k^{\kappa}) \Leftrightarrow \sigma(y|_k^{\kappa})$ for all $x|_k^{\kappa}, y|_k^{\kappa}$ with $k \neq \cdot$;
- τ is a map $\mathcal{P}^{\mathrm{inf}} \to \mathbb{P}^{\mathrm{cnt}}$ such that, for any identifiers $i, j \in \mathcal{I}^{\mathrm{inf}}$: if $\tau(i) = \tau(j)$, then $i = j$.

Apart from the introduction of sets $\mathsf{G}^{\mathrm{ctx}}$ and $\mathcal{G}^{\mathrm{inf}}$ of non-personal information, this definition is the same as the definition of PI Model (Definition 2.1.1) from the previous chapter. Indeed, note that if

$$(\mathsf{P}^{\mathrm{ctx}}, \mathcal{P}^{\mathrm{inf}}, \mathbb{P}^{\mathrm{cnt}}, \Leftrightarrow, \sigma, \tau)$$

is an Information Model, then

$$(\mathsf{O}^{\mathrm{ctx}}, \mathcal{O}^{\mathrm{inf}}, \mathbb{O}^{\mathrm{cnt}}, \Leftrightarrow, \sigma|_{\mathsf{O}^{\mathrm{ctx}}}, \tau|_{\mathcal{O}^{\mathrm{inf}}})$$

is a PI Model.

3

3.2 Model of Cryptographic Messages

We model messages built from personal and non-personal information using cryptographic primitives by means of formal terms. The abstract model we present now is independent from any particular cryptographic primitives; in Section 3.4, we instantiate the model for a set of standard cryptographic primitives including encryption and digital signatures. We model cryptographic primitives by a *signature* Σ consisting of function symbols f with a certain arity[3] k, denoted f/k. For instance, we can represent deterministic symmetric encryption[4] using function symbol $\mathsf{enc}/2$. Given plaintext x and symmetric key y, the encryption of x under key y is then represented as formal term $\mathsf{enc}(x, y)$. More precisely:

[3] I.e., number of parameters

[4] Symmetric encryption is encryption in which they same "symmetric key" is used to encrypt and decrypt the message; in deterministic encryption, the same key and same message always give the same encryption.

Definition 3.2.1. Let $I = (\mathsf{P}^{\mathrm{ctx}}, \mathcal{P}^{\mathrm{inf}}, \mathbb{P}^{\mathrm{cnt}}, \Leftrightarrow, \sigma, \tau)$ be an Information Model, and let Σ be a signature.

- The set $\mathcal{L}^{ctx}$ of *context messages* is the set of formal terms built from context items in $\mathsf{P}^{\mathrm{ctx}}$ by recursive application $f(m_1, \ldots, m_k)$ of function symbols $f/k \in \Sigma$.
- The *submessage* of message $f(m_1, \ldots, m_k)$ at position $p \in \{1, \ldots, k\}$ is denoted $f(m_1, \ldots, m_k)@p := m_p$; the submessage of message m at position $p_1, \ldots, p_m$ is $\mathsf{m}@p_1, \ldots, p_m := ((\mathsf{m}@p_1)\ldots)@p_k$. The *empty position* is denoted ϵ, i.e., $\mathsf{m}@\epsilon = \mathsf{m}$.
- The *content equivalence* relation $\doteq$ on $\mathcal{L}^{ctx}$ is the natural extension of $\doteq$ on $\mathsf{O}^{\mathrm{ctx}}$ (cf. remarks below Definition 2.1.1): for $\mathsf{p}_1, \mathsf{p}_2 \in \mathsf{P}^{\mathrm{ctx}}$, $\mathsf{p}_1 \doteq \mathsf{p}_2$ iff $\tau(\sigma(\mathsf{p}_1)) = \tau(\sigma(\mathsf{p}_2))$; and $f(m_1, \ldots, m_k) \doteq f(n_1, \ldots, n_k)$ iff $m_i \doteq n_i$ for all i.

For instance, given function symbol $\mathsf{enc}/2 \in \Sigma$ and context items $k|_{\cdot}^{\pi} \in \mathsf{P}^{\mathrm{ctx}}$ and $secret|_u^{\pi} \in \mathsf{P}^{\mathrm{ctx}}$, $\mathsf{enc}(secret|_u^{\pi}, k|_{\cdot}^{\pi})$ is a context message that may represent the encryption of the piece of information $secret|_u^{\pi}$ under the key $k|_{\cdot}^{\pi}$. Usually, all context items in a context message have the same domain; for these cases, we introduce a shorthand notation in which we write the domain once at the end of the message. For instance, the above message can be written $\mathsf{enc}(secret|_u, k|_{\cdot})|^{\pi}$.

Our definition of content equivalence reflects two assumptions on message contents: namely, that they are *deterministic* and *unique*.

Namely, the fact that $m_i \doteq n_i$ implies $f(m_1, \ldots, m_k) \doteq f(n_1, \ldots, n_k)$ represents determinism: given the same contents as input, cryptographic primitives always give the same output. Randomness, e.g., in signing or in non-deterministic encryption, can be modelled explicitly as part of the plaintext. By explicitly modelling randomness, we can for instance distinguish the case where an actor observes two different probabilistic encryptions[5] with the same input from the case where he observes the same probabilistic encryption twice; in the latter case, we will allow an actor to draw certain conclusions from this. The reverse implication, i.e., if $f(m_1, \ldots, m_k) \doteq f(n_1, \ldots, n_k)$ then $m_i \doteq n_i$, reflects uniqueness. Note that, a priori, different terms representing different messages could have the same contents; e.g., the hashes of two different values could collide; or the hash of some value could be the same as the encryption of some other value. In our model, this does not happen; hence, we assume that such accidental coincidence of contents does not lead to privacy problems.

[5] In probabilistic encryption, each encryption of the same message with the same key will look different: this way it is impossible to distinguish two encryptions of different values from two encryptions of the same value.

3.3 Deducing Knowledge About Messages

Having described messages containing personal information, we now model what knowledge actors can obtain from them. Namely, we enumerate the operations that actors can perform on messages, and then define what messages an actor can obtain by repeatedly performing these operations. The model in this section is independent from any particular cryptographic primitives; we will instantiate it for standard primitives in Section 3.4. The reasoning in this section is also independent from the structure of context items, i.e., their domain, profile and variable. Abstractly, in this section, we see context items just as a set $\mathsf{P}^{\mathsf{ctx}}$ with a relation $\doteq$ on it, and ask which of these context items can be derived from a set of messages. (We use the structure of context items when applying the results from this section to determine an actor's view in Section 3.5.)

We first describe the two types of operation that an actor can perform on messages: *constructing* new messages, and *eliminating* messages into their parts. We then use a deductive system to describe the repeated application of these operations.

Construction The most basic reasoning step that an actor can make is to build a cryptographic message $\mathsf{f}(x_1, \ldots, x_k)$ from its parts. Which parts are needed depends on the function symbol. Usually, $\mathsf{f}(\mathsf{m}_1, \ldots, \mathsf{m}_k)$ can be constructed if $\mathsf{m}_1, \ldots, \mathsf{m}_k$ are known. However, for particular function symbols this may be different: e.g., we might have a function symbol $\mathsf{dh}/2$ so that $\mathsf{dh}(\mathsf{m}_1, \mathsf{m}_2)$ can be constructed whenever either m_1 and $\mathsf{pk}(\mathsf{m}_2)$, or $\mathsf{pk}(\mathsf{m}_1)$ and m_2 are known[6].

In general, we model the requirements for constructing a message by a *construction rule*. Such a rule is denoted

$$\mathsf{f}(a_1, \ldots, a_k) \leftarrow b_1, \ldots, b_l, \qquad (\circ)$$

[6] Intuitively, this models something similar to Diffie-Hellman key exchange. However, in contrast to real Diffie-Hellman, in this model, $\mathsf{dh}(x, y) \neq \mathsf{dh}(y, x)$ in general.

where a_i, b_i are *variable messages* $\in \mathcal{L}^x$ built from a set $\mathcal{X}$ of *variables*. We call $\mathsf{f}(a_1, \ldots, a_k)$ the *whole message* and b_i the *parts*. For instance, the construction rule for encryption is

$$\mathsf{enc}(x, y) \leftarrow x, y; \qquad \textbf{(enc)}$$

the rules for the above dh example are:

$$\mathsf{dh}(x, y) \leftarrow \mathsf{pk}(x), y \quad \mathsf{dh}(x, y) \leftarrow x, \mathsf{pk}(y).$$

For simplicity, we assume that exactly the same variables occur in the "left-hand-side" $\mathsf{f}(a_1, \ldots, a_k)$ and "right-hand side" $b_1, \ldots, b_l$ of a construction rule; and moreover, that every variable occurs exactly once both left and right. (In Section 3.8, we discuss generalisations.)

The application of a construction rule on actual messages is captured by *instantiating* that rule. Formally, an *instantiation* of construction rule ($\circ$) is a substitution σ of context messages for all variables in the construction rule. In this case, we write

$$\mathsf{f}(a_1\sigma, \ldots, a_k\sigma) \leftarrow b_1\sigma, \ldots, b_l\sigma.$$

We demonstrate instantiations of construction rules in the next example:

Example 3.3.1. Suppose we have context items $k|^{\pi}_{\cdot} \in \mathsf{P}^{\mathsf{ctx}}$, representing a symmetric key, and $secret|^{\pi}_{u} \in \mathsf{P}^{\mathsf{ctx}}$, representing a piece of personal information. Consider construction rule (**enc**) for encryption. We have the following instantiation of this rule:

$$\mathsf{enc}(secret|^{\pi}_{u}, k|^{\pi}_{\cdot}) \leftarrow secret|^{\pi}_{u}, k|^{\pi}_{\cdot};$$

namely, it follows from substitution $\sigma = \{x \to secret|^{\pi}_{u}, y \to k|^{\pi}_{\cdot}\}$. This represents that an actor who knows key $k|^{\pi}_{\cdot}$ and message $secret|^{\pi}_{u}$ can use those to build the given encryption. □

Elimination The operations on cryptographic messages that an actor can perform are modelled by *elimination rules*. Informally, such an elimination rule states that, given a message x of a certain form and given the contents of some additional messages, a part of message x can be recovered by applying the operation. Similarly to construction rules, an *elimination rule* is written

$$\mathsf{f}(a_1, \ldots, a_k) \xrightarrow{\doteq b_1, \ldots, \doteq b_l} c, \qquad (\dagger)$$

where $f/k \in \Sigma$, and $a_i, b_i, c \in \mathcal{L}^x$ are variable messages such that each variable occurs at most once in $f(a_1, \ldots, a_k)$; and each variable occurring in b_i, c also occurs in $f(a_1, \ldots, a_k)$. Here, $f(a_1, \ldots, a_k)$ is the *whole message*; c is the *part*, and b_i are *auxiliary messages*. For instance, consider function symbol enc/2, so that $\mathsf{enc}(x, y)$ represents the encryption of plaintext x under key y. Decryption can be modelled with the following elimination rule:

$$\mathsf{enc}(x, y) \xrightarrow{\doteq y} x. \qquad \textbf{(sdec)}$$

I.e, if an encryption is known and the contents of the corresponding key are known, then the plaintext can be obtained.

Similarly to construction rules, elimination rules are *instantiated* to capture the application of an elimination rule to actual messages. Formally, an *instantiation* of an elimination rule of the form (†) is a substitution σ of context messages for all variables in the elimination rule. If $\mathsf{m} = f(a_1, \ldots, a_k)\sigma$; $\mathsf{m}_i \doteq b_i\sigma$ and $\mathsf{n} = c\sigma$, we write[7]

$$\mathsf{m} \xrightarrow{\mathsf{m}_1,\ldots,\mathsf{m}_l} \mathsf{n}.$$

[7] Note that the substitution defines the auxiliary messages up to content equivalence. Namely, we require $\mathsf{m}_i \doteq b_i\sigma$ instead of $\mathsf{m}_i = b_i\sigma$. This reflects that any message can be used as auxiliary message as long as it has the right contents.

We demonstrate instantiations of elimination rules in the next example:

Example 3.3.2. Consider context message $\mathsf{enc}(secret|_u^\pi, k|_\cdot^\pi)$ representing the encryption of secret $secret|_u^\pi$ under symmetric key $k|_\cdot^\pi$. Let $shakey|_\cdot$ be a context item such that $k|_\cdot^\pi \doteq shakey|_\cdot$, i.e., they have the same contents. We have the following instantiation of decryption rule (**sdec**):

$$\mathsf{enc}(secret|_u^\pi, k|_\cdot^\pi) \xrightarrow{shakey|_\cdot} secret|_u^\pi,$$

namely, it follows from substitution $\sigma = \{x \to secret|_u^\pi, y \to k|_\cdot^\pi\}$.

Intuitively, this means that an actor who knows $\mathsf{enc}(secret|_u^\pi, k|_\cdot^\pi)$ and $shakey|_\cdot$, can determine $secret|_u^\pi$. □

For our purposes, we need to let actors derive more knowledge from a cryptographic operation than just the resulting message. Namely, we model that, if an actor observes that the operation has succeeded, he also knows that the auxiliary messages provided to the operation must have been correct. For instance, in Example 3.3.2, suppose that an actor knows $\mathsf{enc}(secret|_u^\pi, k|_\cdot^\pi)$ and $shakey|_\cdot$. He can perform decryption of the former message using the latter as auxiliary message; if he finds that decryption has succeeded, then he knows that $k|_\cdot^\pi$ and $shakey|_\cdot$ are representations of the same contents, i.e., he now also knows $k|_\cdot^\pi$. Some cryptographic operations, such as signature verification, always return a value indicating whether or not the correct auxiliary information was provided; for others, such as decryption, this depends on the particular implementation. However, even if the operation does not return such a value, actors can usually still see if the correct auxiliary information (e.g., the decryption key) was provided by checking if the result (e.g., the plaintext) looks like random garbage or not.

We call our assumption that actors can check the correctness of auxiliary information the *visible failure* assumption. It may result in an over-approximation of the knowledge of actors; but it considerably simplifies our reasoning model because we do not have to model whether actors can "recognize" the result of an operation. Formally, this assumption means that for any elimination rule

$$\mathsf{f}(a_1, \ldots, a_k) \xrightarrow{\doteq b_1,\ldots,\doteq b_l} c$$

and any $i \in \{1, \ldots, l\}$, there is also an elimination rule

$$\mathsf{f}(a_1, \ldots, a_k) \xrightarrow{\doteq b_1,\ldots,\doteq b_l} b_i.$$

We demonstrate the visible failure assumption in the next example:

Example 3.3.3. Consider elimination rule (**sdec**) above. By visible failure, we also adopt elimination rule

$$\mathsf{enc}(x, y) \xrightarrow{\doteq y} y; \qquad \textbf{(sdec')}$$

so, for instance, an actor who knows $\mathsf{enc}(secret|_u^\pi, k|_\cdot^\pi)$ and $shakey|_\cdot$ such that $k|_\cdot^\pi \doteq shakey|_\cdot$, can apply (**sdec'**) to obtain $k|_\cdot^\pi$. □

Finally, we allow actors to exploit our determinism and uniqueness assumptions on cryptographic primitives to learn new representations of messages. Namely, suppose an actor knows a message $\mathsf{enc}(secret|_u^\pi, k|_\cdot^\pi)$, and he knows its parts $secret|_u^\pi, k|_\cdot^\pi$ in different contexts, say as $data|_{al}^{db} \doteq secret|_u^\pi$ and $shakey|_\cdot \doteq k|_\cdot^\pi$, respectively. Now, due to determinism, encryption $\mathsf{enc}(data|_{al}^{db}, shakey|_\cdot)$ has the same contents as $\mathsf{enc}(secret|_u^\pi, k|_\cdot^\pi)$; due to uniqueness, the actor knows that its parts $data|_{al}^{db}$ and $shakey|_\cdot$ are the same as $secret|_u^\pi$ and $k|_\cdot^\pi$, respectively. To allow this kind of reasoning, we introduce *reconstruction rules*: formally, for every construction rule $\mathsf{f}(a_1, \ldots, a_k) \leftarrow b_1, \ldots, b_l$ and every $i \in \{1, \ldots, l\}$, we have a corresponding elimination rule

$$\mathsf{f}(a_1, \ldots, a_k) \xrightarrow{\doteq b_1, \ldots, \doteq b_l} b_i \in \mathcal{E}.$$

For instance, in the above case, this rule for enc allows us to derive $secret|_u^\pi$ and $k|_\cdot^\pi$ given $data|_{al}^{db}$ and $shakey|_\cdot$[8].

Elimination rules due to visible failure and reconstruction are examples of what we call *testing rules*. Namely, testing rules are elimination rules $\mathsf{f}(a_1, \ldots, a_k) \xrightarrow{\doteq b_1, \ldots, \doteq b_l} c$ for which c coincides with one of the b_i. Such rules serve purely to derive new representations of messages whose contents were already known; they will have a special role in the algorithms we develop in Section 3.7[9].

Deductive System Having discussed all single reasoning steps that an actor can apply, we now introduce a deductive system that captures which messages an actor can obtain by repeatedly applying them. Below, assume that a signature with an associated set of construction and elimination rules is fixed. Given this signature and rules, the deductive system will give the semantics of the relation $\mathcal{C} \vdash \mathsf{m}$, where $\mathcal{C} \subset \mathcal{L}^{ctx}$ is a set of context messages, and m is a context message. Here, $\mathcal{C} \vdash \mathsf{m}$ means that can derive message m by applying cryptographic operations to messages in $\mathcal{C}$. Formally:

Definition 3.3.4. Let $\mathcal{C} \subset \mathcal{L}^{ctx}$ be a set of context messages, and m a context message. We say that m is *derivable* from $\mathcal{C}$, denoted $\mathcal{C} \vdash \mathsf{m}$, if the conclusion $\mathcal{C} \vdash \mathsf{m}$ follows from the deductive system[10] in Figure 3.1.

Rule ($\vdash$**0**) states that any known message can be derived. Rule ($\vdash$**C**) states that any instantiation of a construction rule can be used to build a cryptographic primitive from its parts. Rule ($\vdash$**E**) states

[8] In this case, the reconstruction rules are redundant because (**sdec**) and (**sdec'**) can be used to derive the same parts with fewer auxiliary messages. In general, this will not be the case.

[9] There may also be testing rules that do not come from the two above assumptions; cf. our model of anonymous credentials in Section 6.5.

[10] A deductive system is a set of inference rules. Each rule states that if a set of premises is satisfied, then the conclusion follows. Such a rule is denoted

$$\frac{p_1 \quad \ldots \quad p_k}{q}\,(C)\ \ (N),$$

where the p_i are the premises, q is the conclusion, C is a condition under which the rule may be applied, and N is the name of the rule. A statement "follows" from the deductive system by repeated application of these rules. Such reasoning can be visualised as a tree (e.g., Figure 3.2): each node denotes the application of an inference rule, leaf nodes are axioms (rules without premises), and the root is the conclusion.

$$\frac{}{\mathcal{C} \vdash \mathsf{m}}\,(\mathsf{m} \in \mathcal{C})\ (\vdash\mathbf{0}) \qquad \frac{\mathcal{C} \vdash \mathsf{m}_1 \quad \ldots \quad \mathcal{C} \vdash \mathsf{m}_l}{\mathcal{C} \vdash f(\mathsf{n}_1, \ldots, \mathsf{n}_k)}\,(f(\mathsf{n}_1, \ldots, \mathsf{n}_k) \leftarrow \mathsf{m}_1, \ldots, \mathsf{m}_l)\ (\vdash\mathbf{C})$$

$$\frac{\mathcal{C} \vdash \mathsf{m} \quad \mathcal{C} \vdash \mathsf{m}_1 \quad \ldots \quad \mathcal{C} \vdash \mathsf{m}_k}{\mathcal{C} \vdash \mathsf{n}}\,(\mathsf{m} \xrightarrow{\mathsf{m}_1, \ldots, \mathsf{m}_k} \mathsf{n})\ (\vdash\mathbf{E})$$

Figure 3.1: Inference rules for message derivability ($\mathsf{m}, \mathsf{m}_i, \mathsf{n}, \mathsf{n}_i$ any context messages; $f/k \in \Sigma$)

that any instantiation of an elimination rule can be applied to known messages.

The next example demonstrates derivability:

Example 3.3.5. Consider function symbol enc and rule (**sdec**) as above. Given $\mathcal{C} = \{\mathsf{enc}(\mathit{secret}|_u^\pi, k|_\cdot^\pi), \mathit{shakey}|_\cdot\}$, we have that $\mathcal{C} \vdash \mathit{secret}|_u^\pi$. Informally, the encryption and decryption key are known (translating to an application of inference rule ($\vdash\mathbf{0}$)), so the plaintext can be recovered by decryption (inference rule ($\vdash\mathbf{E}$) with $\mathsf{enc}(\mathit{secret}|_u^\pi, k|_\cdot^\pi) \xrightarrow{\mathit{shakey}|_\cdot} \mathit{secret}|_u^\pi$). This reasoning is formalised in Figure 3.2. □

3.4 Modelling Standard Cryptographic Primitives

We now show how standard (cryptographic) primitives can be modelled using the above formalism. We model concatenation, symmetric encryption, asymmetric encryption, cryptographic hashes and digital signatures. Our model of these primitives is deterministic, i.e., given the same inputs, the cryptographic primitives always output the same contents. We discuss the modelling of probabilistic promitives (that output different contents) at the end of this section.

Concatenation Because concatenation of messages is a very common operation, we introduce a special syntax for it. Namely, instead using a function symbol, we denote a k-length list as $\{x_1, \ldots, x_k\}$. It satisfies the following construction and elimination rules[11]:

$$\{x_1, \ldots, x_k\}{\leftarrow}x_1, \ldots, x_k; \qquad \{x_1, \ldots, x_k\}{\rightarrow}x_i.$$

[11] Testing rules $\{x_1, \ldots, x_k\} \xrightarrow{\doteq x1, \ldots, \doteq x_k} x_i$ due to reconstruction are redundant because the same parts can be derived with no auxiliary messages. In the remainder of this section, we leave redundant testing rules implicit.

Note that our model of concatenation is not associative: e.g., $\{x, \{y, z\}\}$ and $\{\{x, y\}, z\}$ are non-content-equivalent. If this behaviour is not desired, then nested lists should be avoided and representation $\{x, y, z\}$ used instead.

In the remainder of this thesis, we assume that any signature includes concatenation.

$$\frac{\dfrac{}{\mathcal{C} \vdash \mathsf{enc}(\mathit{secret}|_u^\pi, k|_\cdot^\pi)}\,(\vdash\mathbf{0}) \qquad \dfrac{}{\mathcal{C} \vdash \mathit{shakey}|_\cdot}\,(\vdash\mathbf{0})}{\mathcal{C} \vdash \mathit{secret}|_u^\pi}\,(\vdash\mathbf{E})$$

Figure 3.2: Derivation of $\mathit{secret}|_u^\pi$ from $\mathcal{C}$ (Example 3.3.5)

Symmetric Encryption As discussed above, symmetric encryption is encryption in which the same key is used to encrypt and decrypt the message. Symmetric encryption can be modelled with function symbol $\mathsf{enc}/2$, so that $\mathsf{enc}(x, y)$ represents the encryption of message x under key y. The construction rule for enc represents encryption. The first elimination rule represents decryption; the second elimination rule represents the ability of an actor to verify if a given key was used (cf. Example 3.3.3):

$$\mathsf{enc}(x, y) \leftarrow x, y; \qquad \mathsf{enc}(x, y) \xrightarrow{\doteq y} x; \qquad \mathsf{enc}(x, y) \xrightarrow{\doteq y} y.$$

3

Asymmetric Encryption In asymmetric encryption, keys come in pairs consisting of a private key and a public key: the public key (which is freely distributed) is used to encrypt messages that only the holder of the private key (which should be kept secret) can decrypt.

We model private/public key pairs using a function symbol $\mathsf{pk}/1$: $\mathsf{pk}(x)$ represents the public key corresponding to private key x. It has construction rule $\mathsf{pk}(x) \leftarrow x$. Hence, in this model, the public key can be derived from the private key but not the other way around. However, a private key can be verified to correspond to a public key using reconstruction rule $\mathsf{pk}(x) \xrightarrow{\doteq x} x$.

Then, asymmetric encryption can be modelled with a function symbol $\mathsf{aenc}/2$: $\mathsf{aenc}(x, \mathsf{pk}(y))$ represents the encryption of message x using public key $\mathsf{pk}(y)$. It has construction rule $\mathsf{aenc}(x, y) \leftarrow x, y$[12]. Decryption is modelled by rule

[12] Or $\mathsf{aenc}(x, \mathsf{pk}(y)) \leftarrow x, \mathsf{pk}(y)$: this makes no practical difference.

$$\mathsf{aenc}(x, \mathsf{pk}(y)) \xrightarrow{\doteq y} x;$$

the corresponding rule[13] to verify if a decryption key is correct, is

[13] By the visible failure assumption

$$\mathsf{aenc}(x, \mathsf{pk}(y)) \xrightarrow{\doteq y} y.$$

Moreover, the reconstruction rules

$$\mathsf{aenc}(x, \mathsf{pk}(y)) \xrightarrow{\doteq x, \doteq \mathsf{pk}(y)} x, \qquad \mathsf{aenc}(x, \mathsf{pk}(y)) \xrightarrow{\doteq x, \doteq \mathsf{pk}(y)} \mathsf{pk}(y)$$

capture the possibility to re-perform an encryption and compare results. (Note that in practice, cryptographic protocols almost always use probabilistic asymmetric encryption as described below.)

Cryptographic Hash Functions Intuitively, cryptographic hash functions are functions that are easy to compute, but hard to invert.[14] We model this simply by introducing a function symbol $\mathsf{h}/1$ with construction rule $\mathsf{h}(x) \leftarrow x$ but *without any elimination rules* other than the obligatory reconstruction rule $\mathsf{h}(x) \xrightarrow{\doteq x} x$ (which models that, given a hashed value and its original, we can re-compute the hash to observe their correspondence).

[14] More precisely, given a hash, it is hard to find a message with that hash; given a message, it is hard to find another message with the same hash; and it is hard to find two different messages with the same hash; cf. Menezes et al. (1996)

Digital Signatures A digital signature demonstrates the authenticity of a message. Namely, a digital signature on a message can be

verified using a public key: if this verification succeeds, this guarantees that the message was originally signed using the corresponding private key (and not modified afterwards). Because of applications of our model later in this thesis, we model digital signatures "with message recovery"[15]: that is, the message itself can be recovered from the signature.

[15] Menezes et al. (1996)

Digital signatures are modelled by function symbol $\mathsf{sig}/2$: $\mathsf{sig}(x, y)$ represents the digital signature on message x signed with private key y. Its construction rule is $\mathsf{sig}(x, y) \leftarrow x, y$. Verification of a digital signature is modelled by elimination rules[16]:

$$\mathsf{sig}(x, y) \xrightarrow{\div \mathsf{pk}(y)} \mathsf{x}; \qquad \mathsf{sig}(x, y) \xrightarrow{\div \mathsf{pk}(y)} \mathsf{pk}(y).$$

Hence, if verification succeeds, then the actor knows that it was performed with the right public key, and he obtains the message signed.

[16] In this case, we don't give reconstruction rules because they are redundant: the message can be derived using just the public key, and the private key can be tested by first testing the public key $\mathsf{pk}(y)$ and then testing the private key from $\mathsf{pk}(y)$.

Digital signatures "with appendix"[17], i.e., for which the message is needed for verification, are modelled straightforwardly by adding $\div \mathsf{x}$ to the auxiliary messages for the above elimination rules.

[17] Menezes et al. (1996)

Probabilistic Cryptographic Primitives Above, we modelled deterministic cryptographic primitives. In practice, usually probabilistic primitives are used. Two deterministic encryptions of the same message have the same contents; hence even if an attacker does not know the decryption key, he still knows that they contain the same plaintext. Probabilistic encryption prevents this by using randomness in the encryption process; similarly for other primitives. We can model probabilistic cryptographic primitives by adding this randomness explicitly to the plaintext. For instance, we model a probabilistic symmetric encryption of message $secret|_k^\pi$ using key $key|^\pi$ as $\mathsf{enc}(\{secret|_k^\pi, n|^\pi\}, key|^\pi)$, where $n|^\pi$ represents the randomness. Probabilistic asymmetric encryption and probabilistic digital signatures are modelled analogously[18].

[18] Alternatively, a 3-ary function symbol can be used (cf. Blanchet and Smyth (2011)): our model differs from that latter model in that our decryption operation also returns the randomness.

Notation In the remainder of this chapter, for our examples we will use the above signature

$$\Sigma = \{\mathsf{enc}/2, \mathsf{pk}/1, \mathsf{aenc}/2, \mathsf{h}/1, \mathsf{sig}/2\}$$

(plus concatenation) and associated construction/elimination rules.

3.5 View from a Knowledge Base

Having discussed what messages an actor can obtain by applying cryptographic operations, we now obtain an actor's view by interpreting these messages as personal information. Namely, let us model the complete knowledge of an actor as a finite set[19] $\mathcal{C} \subset \mathcal{L}^{ctx}$ of context messages called his *knowledge base*. In addition to the messages the actor has sent and received, this knowledge base should also contain any additional personal information we want to reason

[19] In other approaches, the messages known by an actor are typically represented by an ordered sequence. For our purposes, a set is sufficient: intuitively, the "order" of messages is captured by the different context-layer representations of information.

about, such as information from databases held by the actor. Also, the knowledge base should contain relevant material like secret keys known by the actor, and nonces he has generated during the execution of the cryptographic protocols. The objective of this section is to derive the view of an actor (cf. Definition 2.2.1) given his knowledge base.

The following example demonstrates knowledge bases.

Example 3.5.1. We consider the PI Model of Example 2.1.2. We give knowledge bases of the client and server for an example communication protocol. In the first message, the client sends a symmetric encryption of Alice's identifier to the server, encrypted using a shared key. We model the shared key by a non-personal item with context-layer representation $shkey|_{\cdot}^{\pi}$. The encryption is then modelled by

$$\mathsf{m}_1 = \mathsf{enc}(id|_{su}, shkey|_{\cdot})|^{\pi}.$$

In the second message, the server replies with an encryption under $shkey|_{\cdot}^{\pi}$ containing both Alice's identifier and a probabilistic signature on her age using the server's secret key. The randomness used in the signature is represented as a non-personal item with context-layer representation $n|_{\cdot}^{\pi}$. The secret key of the server is context identifier $k^{-}|_{srv}^{\pi}$. The second message is:

$$\mathsf{m}_2 = \mathsf{enc}(\{id|_{su}, \mathsf{sig}(\{age|_{su}, n|_{\cdot}\}, k^{-}|_{srv})\}, shkey|_{\cdot})|^{\pi}.$$

We now consider the knowledge base of the client. The client knows $\mathsf{m}_1, \mathsf{m}_2$. In addition, his knowledge base contains the personal and other information he knew beforehand. Apart from his address book, we assume that this initial knowledge includes the shared key and the public key of the server, known in some arbitrary contexts $*|_{\cdot}^{\cdot}$, $*|_{srv}^{\cdot}$. His full knowledge base after communication is then:

$$\mathcal{C}_{cli} = \{nm|_{12}^{ab}, teln|_{12}^{ab}, nm|_{4}^{ab}, id|_{4}^{ab}, skey|_{\cdot}^{\cdot}, \mathsf{pk}(k^{-}|_{srv}^{\cdot}), \mathsf{m}_1, \mathsf{m}_2\}.$$

Similarly, the knowledge base of the server can be modelled:

$$\mathcal{C}_{srv} = \{col1|_{1}^{db}, key|_{1}^{db}, col1|_{2}^{db}, key|_{2}^{db}, n|_{\cdot}^{\cdot}, skey|_{\cdot}^{\cdot}, k^{-}|_{srv}^{\cdot}, \mathsf{m}_1, \mathsf{m}_2\}$$

where $n|_{\cdot}^{\cdot} \doteq n|_{\cdot}^{\pi}$ is the nonce the server uses in his answer. □

Given knowledge bases $\{\mathcal{C}_a\}_{a\in\mathcal{A}}$ of a set of actors $\mathcal{A}$, the knowledge base of the coalition $\mathcal{A}$ of actors is taken to be the union of the knowledge bases of the respective actors. For instance, in the above example, the knowledge base of coalition $\{cli, srv\}$ is $\mathcal{C}_{cli} \cup \mathcal{C}_{srv}$.

Our procedure to derive a view from a knowledge base is based on the following two main assumptions:

- *Detecting from messages* — An actor learns a piece of information by reading it directly from a message, or by determining that its contents are the same as that of some information he already knows.

- *Associating by identifiers* — An actor associates two contexts by determining that an identifier with the same contents occurs in both of them.

These assumptions reflect our focus on analysing protocols for transmitting personal information (i.e.., what can be derived because of the message formats that are used) rather than personal information itself (i.e., what could be derived by interpreting this information, e.g., by deriving a zip code from an address). We discuss these assumptions in Section 3.8.

The above assumptions imply that, apart from knowledge derivable from messages using cryptographic operations, we also need to consider knowledge derivable from observing that different messages represent the same contents. In more detail, suppose an actor knows two deterministic encryptions $\mathsf{enc}(\mathsf{i}_1, \mathsf{k})$, $\mathsf{enc}(\mathsf{i}_2, \mathsf{k})$, where $\mathsf{i}_1 \doteq \mathsf{i}_2$, so also $\mathsf{enc}(\mathsf{i}_1, \mathsf{k}) \doteq \mathsf{enc}(\mathsf{i}_2, \mathsf{k})$. Now, even if the actor cannot derive i_1 or i_2, because of our uniqueness assumption he still knows that the two plaintexts must have the same contents. This has two consequences. First, if he happens to know i_1 from somewhere else, e.g. if $\mathcal{C} = \{\mathsf{enc}(\mathsf{i}_1, \mathsf{k}), \mathsf{enc}(\mathsf{i}_2, \mathsf{k}), \mathsf{i}_1\}$, then by the above reasoning, he also knows i_2. Second, if i_1 and i_2 represent identifiers, then he knows that the contexts that they are in must both represent the same data subject. We formalise such knowledge as follows:

Definition 3.5.2. Let $\mathcal{C} \subset \mathcal{L}^{ctx}$ be a set of context messages, and $\mathsf{p}_1, \mathsf{p}_2 \in \mathsf{P}$ two context items.

- We say that $\mathsf{p}_1, \mathsf{p}_2$ are *directly equatable* from $\mathcal{C}$, denoted $\mathcal{C} \vdash \mathsf{p}_1 \doteq_0 \mathsf{p}_2$, if there are $\mathsf{m}_1, \mathsf{m}_2, z$ such that $\mathcal{C} \vdash \mathsf{m}_1$, $\mathcal{C} \vdash \mathsf{m}_2$, $\mathsf{m}_1 \doteq \mathsf{m}_2$, $\mathsf{m}_1@z = \mathsf{p}_1$, and $\mathsf{m}_2@z = \mathsf{p}_2$.
- We write the transitive closure of $\mathcal{C} \vdash * \doteq_0 *$ as $\mathcal{C} \vdash * \doteq *$; if $\mathcal{C} \vdash \mathsf{p}_1 \doteq \mathsf{p}_2$, then $\mathsf{p}_1, \mathsf{p}_2$ are *equatable* from $\mathcal{C}$.

We now define the view of an actor, i.e., the knowledge of personal information that the actor can derive from his knowledge base, taking the above considerations into account:

Definition 3.5.3. Let $\mathcal{C} \subset \mathcal{L}^{ctx}$ be a set of context messages. The *view corresponding to* $\mathcal{C}$ is the view $V = (\mathsf{O}, \leftrightarrow)$ such that:

- The set $\mathsf{O} \subset \mathsf{P}^{\mathsf{ctx}}$ of detectable items is the set of all context personal items that are derivable from $\mathcal{C}$ (Definition 3.3.4) or equatable from $\mathcal{C}$ to a context item derivable from $\mathcal{C}$ (Definition 3.5.2);
- The associability relation $\leftrightarrow$ is the least equivalence relation on contexts in $\mathsf{O}^{\mathsf{ctx}}$ such that, whenever $\mathcal{C} \vdash i|_k^\pi \doteq i'|_l^\eta$ for context identifiers $i|_k^\pi, i'|_l^\eta$, then $*|_k^\pi \leftrightarrow *|_l^\eta$.[20]

The following examples demonstrate the definition.

[20] Note that direct equatability is not sufficient because the "intermediate" context items may not be identifiers.

Example 3.5.4. Consider the knowledge base $\mathcal{C}_{cli}$ from Example 3.5.1:

$$\mathcal{C}_{cli} = \{nm|_{12}^{ab}, teln|_{12}^{ab}, nm|_4^{ab}, id|_4^{ab}, skey|_{\cdot}^{\cdot}, \mathsf{pk}(k^-|_{srv}^{\cdot}), \mathsf{enc}(id|_{su}, shkey|_{\cdot}^{\cdot})|^\pi, \\ \mathsf{enc}(\{id|_{su}, \mathsf{sig}(\{age|_{su}, n|_{\cdot}^{\cdot}\}, k^-|_{srv})\}, shkey|_{\cdot}^{\cdot})|^\pi\}.$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathcal{C}_{cli} \vdash \mathsf{enc}(\{id|_{su}, \ldots\}, shkey|_{\cdot})|^{\pi}}(\vdash\mathbf{0}) \quad \dfrac{}{\mathcal{C}_{cli} \vdash skey|^{\cdot}_{\cdot}}(\vdash\mathbf{0})}{\mathcal{C}_{cli} \vdash \{age|_{su}, n|_{\cdot}, \mathsf{sig}(\{age|_{su}, n|_{\cdot}\}, k^{-}|_{srv})\}|^{\pi}}(\vdash\mathbf{E})}{\mathcal{C}_{cli} \vdash \mathsf{sig}(\{age|_{su}, n|_{\cdot}\}, k^{-}|_{srv})|^{\pi}}(\vdash\mathbf{E}) \quad \dfrac{}{\mathcal{C}_{cli} \vdash \mathsf{pk}(k^{-}|^{\cdot}_{srv})}(\vdash\mathbf{0})}{\mathcal{C}_{cli} \vdash \mathsf{pk}(k^{-}|^{\pi}_{srv})}(\vdash\mathbf{E})$$

Figure 3.3: Derivation of the server's public key from the client's knowledge base (Example 3.5.4)

Let us now consider the view $V_{cli} = (\mathsf{O}_{cli}, \leftrightarrow_{cli})$ corresponding to this knowledge base. It is not hard to see that the context items derivable from $\mathcal{C}_{cli}$ are exactly the set

$$\{nm|^{ab}_{12}, teln|^{ab}_{12}, nm|^{ab}_{4}, id|^{ab}_{4}, id|^{\pi}_{su}, age|^{\pi}_{su}\}.$$

In particular, the only pieces of information that are non-derivable are $k^{-}|^{\pi}_{srv}$ and $k^{-}|^{\cdot}_{srv}$; also, both are not content equivalent to any derivable context item, so they cannot be detectable. Therefore:

$$\mathsf{O}_{cli} = \{nm|^{ab}_{12}, teln|^{ab}_{12}, nm|^{ab}_{4}, id|^{ab}_{4}, id|^{\pi}_{su}, age|^{\pi}_{su}\}.$$

Concerning the associability relation $\leftrightarrow_{cli}$, we note that $id|^{ab}_{4} \doteq id|^{\pi}_{su}$ and both are derivable, whence $\mathcal{C}_{cli} \vdash id|^{ab}_{4} \doteq id|^{\pi}_{su}$. This implies $*|^{ab}_{4} \leftrightarrow_{cli} *|^{\pi}_{su}$. Also, note that $\mathcal{C}_{cli} \vdash \mathsf{pk}(k^{-}|^{\cdot}_{srv})$ and $\mathcal{C}_{cli} \vdash \mathsf{pk}(k^{-}|^{\pi}_{srv})$ (see Figure 3.3 for the latter derivation). Hence $\mathcal{C}_{cli} \vdash k^{-}|^{\cdot}_{srv} \doteq k^{-}|^{\pi}_{srv}$, so $*|^{\cdot}_{srv} \leftrightarrow_{cli} *|^{\pi}_{srv}$ (despite the fact that the client does not know the private keys).

In fact, the view of the client is as shown in Figure 2.4, except that the figure does not show associability of contexts $*|^{\cdot}_{srv}$ and $*|^{\pi}_{srv}$ about the server. □

Example 3.5.5. As in the above example, the view of the server corresponding to knowledge base $\mathcal{C}_{srv}$ from Example 3.5.1, is as in Figure 2.4 apart from information about the server itself. Similarly, also the view of the coalition of client and server in Figure 2.4 is the view corresponding to the coalition's knowledge base $\mathcal{C}_{cli} \cup \mathcal{C}_{srv}$. □

3.6 An Alternative Deductive System

In this section, we present an alternative deductive system that, under reasonable assumptions on the construction and elimination rules used, computes the same knowledge about personal information as the original one. As we will explain, this alternative deductive system is easier to evaluate automatically; it also serves as basis of our symbolic model in Chapter 5.

The main idea behind the alternative deductive system is that it is usually unnecessary to consider the application of elimination rules to constructed messages. Namely, the same result can usually also be obtained from the parts that the message was constructed from. Hence, we define a deductive system that does not allow

$$\frac{}{\mathcal{C}\vdash^{-}\mathsf{m}}\,(\mathsf{m}\in\mathcal{C})\;(\vdash^{-}\mathbf{0}) \qquad \frac{\mathcal{C}\vdash^{-}\mathsf{m}\quad\mathcal{C}\vdash^{+}\mathsf{m}_1\quad\dots\quad\mathcal{C}\vdash^{+}\mathsf{m}_k}{\mathcal{C}\vdash^{-}\mathsf{n}}\,(\mathsf{m}\xrightarrow{\mathsf{m}_1,\dots,\mathsf{m}_k}\mathsf{n})\;(\vdash^{-}\mathbf{E})$$

$$\frac{\mathcal{C}\vdash^{-}\mathsf{m}}{\mathcal{C}\vdash^{+}\mathsf{m}}\,(\vdash^{+}\mathbf{0}) \qquad \frac{\mathcal{C}\vdash^{+}\mathsf{n}_1\quad\dots\quad\mathcal{C}\vdash^{+}\mathsf{n}_l}{\mathcal{C}\vdash^{+}f(\mathsf{m}_1,\dots,\mathsf{m}_k)}\,(\mathsf{f}(\mathsf{m}_1,\dots,\mathsf{m}_k)\leftarrow\mathsf{n}_1,\dots,\mathsf{n}_l)\;(\vdash^{+}\mathbf{C})$$

Figure 3.4: Alternative inference rules for message derivability ($\mathsf{m}, \mathsf{m}_i, \mathsf{n}$ any context messages)

elimination after construction. In this alternative deductive system, it is easier to determine if a message m can be derived because the computation can be split into two parts. First, we compute the set $\mathcal{C}^{\infty}$ of all messages that can be obtained by applying elimination rules on messages in $\mathcal{C}$. Next, we check if m can be obtained from messages in $\mathcal{C}^{\infty}$ using construction rules.

We define the alternative deductive system in Figure 3.4. Namely, we write $\mathcal{C}\vdash^{-}\mathsf{m}$ if message m can be derived from knowledge base $\mathcal{C}$ using elimination rules. We write $\mathcal{C}\vdash^{+}\mathsf{m}$ if message m can be derived by applying construction rules to messages derived using $\vdash^{-}$. Note that the definition of $\vdash^{-}$ depends on that of $\vdash^{+}$, so that the auxiliary messages used for elimination rules can be constructed; similarly, the definition of $\vdash^{+}$ depends on that of $\vdash^{-}$.

We formalise the intuition that elimination after construction is "usually not necessary" by the following assumption:

Assumption 3.6.1 (EC)**.** Let $\mathcal{C}$ be a set of context messages, and m a context message such that $\mathcal{C}\vdash\mathsf{m}$. Then there exists a derivation for $\mathcal{C}\vdash\mathsf{m}$ in the deductive system of Figure 3.1 such that, for any application

$$\frac{\mathcal{C}\vdash\mathsf{f}(\mathsf{m}_1,\dots,\mathsf{m}_k)\quad\mathcal{C}\vdash\mathsf{n}_1\quad\dots\quad\mathcal{C}\vdash\mathsf{n}_l}{\mathcal{C}\vdash\mathsf{n}}\,(\vdash\mathbf{E})$$

of rule $(\vdash\mathbf{E})$ in the derivation, the premise $\mathcal{C}\vdash\mathsf{f}(\mathsf{m}_1,\dots,\mathsf{m}_k)$ is not derived using a construction rule.

Under this assumption, the alternative deductive system is indeed correct:

Proposition 3.6.2. For any set of inference rules that satisfies EC, any set of context messages $\mathcal{C}$ and any context message m:

$$\mathcal{C}\vdash\mathsf{m}\ \text{ iff }\ \mathcal{C}\vdash^{+}\mathsf{m}.$$

Proof. ($\Leftarrow$) Any deduction using $\vdash^{+}$ can be translated into a deduction using $\vdash$ by replacing all applications of $(\vdash^{-}\mathbf{0})$ by $(\vdash\mathbf{0})$; of $(\vdash^{+}\mathbf{C})$ by $(\vdash\mathbf{C})$; and of $(\vdash^{-}\mathbf{E})$ by $(\vdash\mathbf{E})$; and replacing applications of $(\vdash^{+}\mathbf{0})$ by their premise.

($\Rightarrow$) It follows by induction that any deduction for m with $(\vdash\mathbf{0})$ or $(\vdash\mathbf{E})$ as root node can be translated into a deduction of $\mathcal{C}\vdash^{-}\mathsf{m}$; and any deduction with $(\vdash\mathbf{C})$ as root node into a deduction of $\mathcal{C}\vdash^{+}\mathsf{m}$: by the EC assumption, the leftmost child of an application of $(\vdash\mathbf{E})$ is a $(\vdash\mathbf{0})$- or $(\vdash\mathbf{E})$-node. □

To derive personal information, we can clearly use $\vdash^-$ instead of $\vdash^+$:

Corollary 3.6.3. For any set of inference rules that satisfies EC, any set of context messages $\mathcal{C}$ and any context item p:

$$\mathcal{C} \vdash \mathsf{p} \text{ iff } \mathcal{C} \vdash^- \mathsf{p}.$$

Proof. Use the previous proof: in the derivation of $\mathcal{C} \vdash \mathsf{p}$, the last step is always a ($\vdash$**0**)- or ($\vdash$**E**)-node. □

For most sets of construction/elimination rules, $\vdash^-$ is also sufficient for equatability. Recall that for direct equatability of $\mathsf{p}_1, \mathsf{p}_2$, we need that an actor can derive two content equivalent messages $\mathsf{m}_1, \mathsf{m}_2$ in which p_1 and p_2 occur at the same location (cf. Definition 3.5.2). In general, the derivations of messages m_1 and m_2 might both end with an application of rule ($\vdash$**C**). The following assumption asserts that, for many primitives, it is unnecessary to consider the case when both derivations end with ($\vdash$**C**):

Assumption 3.6.4 (EE)**.** Suppose that $\mathcal{C} \vdash \mathsf{p}_1 \doteq_0 \mathsf{p}_2$. Then there exist $\mathsf{m}_1, \mathsf{m}_2, z$ such that $\mathcal{C} \vdash \mathsf{m}_1$, $\mathcal{C} \vdash \mathsf{m}_2$, $\mathsf{m}_1@z = \mathsf{p}_1$, $\mathsf{m}_2@z = \mathsf{p}_2$, and either m_1 or m_2 can be derived using a derivation that does not have ($\vdash$**C**) as root node.

If EC and EE are both satisfied, then $\vdash^-$ is sufficient for equatability:

Proposition 3.6.5. For any set of inference rules that satisfies EC and EE, any set of context messages $\mathcal{C}$ and any pair $\mathsf{p}_1, \mathsf{p}_2$ of context items: $\mathcal{C} \vdash \mathsf{p}_1 \doteq_0 \mathsf{p}_2$ if and only if there are $\mathsf{m}_1, \mathsf{m}_2, z$ such that $\mathcal{C} \vdash^- \mathsf{m}_1$, $\mathcal{C} \vdash^- \mathsf{m}_2$, $\mathsf{m}_1 \doteq \mathsf{m}_2$, $\mathsf{m}_1@z = \mathsf{p}_1$, and $\mathsf{m}_2@z = \mathsf{p}_2$.

Proof. Let $\mathcal{C}, \mathsf{p}_1, \mathsf{p}_2$ be given. By the EE assumption, there are $\mathsf{m}_1, \mathsf{m}_2, z$ such that $\mathcal{C} \vdash \mathsf{m}_1$, $\mathcal{C} \vdash \mathsf{m}_2$, $\mathsf{m}_1@z = \mathsf{p}_1$, $\mathsf{m}_2@z = \mathsf{p}_2$, and not both are obtained by construction rules. Let us say that m_1 is not obtained by a construction rule (the other case is analogous). By the EC assumption, in fact $\mathcal{C} \vdash^- \mathsf{m}_1$.

If m_2 is also not obtained by construction rule, then the proposition holds. Let us instead assume that it is: we have construction rule

$$\mathsf{f}(a_1, \ldots, a_k) \leftarrow b_1, \ldots, b_l$$

and substitution σ such that $\mathsf{m}_2 = \mathsf{f}(a_1, \ldots, a_k)\sigma$ and $\mathcal{C} \vdash b_i\sigma$ for $i \in \{1, \ldots, l\}$. Because each variable in $\mathsf{f}(a_1, \ldots, a_k)$ occurs exactly once in $b_1, \ldots, b_l$, there is a $q \in \{1, \ldots, l\}$ and location z' in b_q such that $\mathsf{m}_2@z = b_q\sigma@z'$. Moreover, if σ' is the substitution such that $\mathsf{f}(a_1, \ldots, a_k)\sigma' = \mathsf{m}_1$, then also $b_q\sigma'@z' = \mathsf{p}_1$. Now, apply the reconstruction rule for this construction rule to message m_1:

$$\frac{\mathcal{C} \vdash^- \mathsf{m}_1 \quad \mathcal{C} \vdash^+ b_1\sigma \quad \ldots \quad \mathcal{C} \vdash^+ b_l\sigma}{\mathcal{C} \vdash^- b_q\sigma'} (\vdash^-\mathbf{E})$$

Hence, we have found messages $b_q\sigma'$, $b_q\sigma$ such that $b_q\sigma'@z' = \mathsf{p}_1$, $b_q\sigma@z' = \mathsf{p}_2$, $\mathcal{C} \vdash^{-} b_q\sigma'$, and $\mathcal{C} \vdash b_q\sigma$. If the derivation of $b_q\sigma$ does not have ($\vdash$**C**) as root node, then by the EC assumption, $\mathcal{C} \vdash^{-} b_q\sigma$, so we are done taking $\mathsf{m}_1 = b_q\sigma'$ and $\mathsf{m}_2 = b_q\sigma$. Otherwise, repeat the above procedure. □

The EC and EE assumptions hold for all models of primitives we consider in this thesis; and their validity can easily be checked for other models. One way to verify the EC assumption, is to show that any partial derivation tree for $\mathcal{C} \vdash \mathsf{n}$ of the form

$$\dfrac{\dfrac{\mathcal{C} \vdash \mathsf{m}'_1 \quad \ldots \quad \mathcal{C} \vdash \mathsf{m}'_q}{\mathcal{C} \vdash \mathsf{f}(\mathsf{m}_1, \ldots, \mathsf{m}_k)}(\vdash\mathbf{C}) \qquad \mathcal{C} \vdash \mathsf{n}_1 \quad \ldots \quad \mathcal{C} \vdash \mathsf{n}_l}{\mathcal{C} \vdash \mathsf{n}}(\vdash\mathbf{E})$$

can be turned into a partial derivation tree for $\mathcal{C} \vdash \mathsf{n}$ with strictly fewer nodes. For instance, consider construction and elimination rules

$$\mathsf{aenc}(x, y) \leftarrow x, y \qquad \mathsf{aenc}(x, \mathsf{pk}(y)) \xrightarrow{\doteq y} y$$

for asymmetric encryption. Supppose that m_2 can be derived with the following partial derivation tree (for some $\mathsf{m}_1, \mathsf{n}_2$):

$$\dfrac{\dfrac{\mathcal{C} \vdash \mathsf{m}_1 \quad \mathcal{C} \vdash \mathsf{pk}(\mathsf{m}_2)}{\mathcal{C} \vdash \mathsf{aenc}(\mathsf{m}_1, \mathsf{pk}(\mathsf{m}_2))}(\vdash\mathbf{C}) \qquad \mathcal{C} \vdash \mathsf{n}_2}{\mathcal{C} \vdash \mathsf{m}_2}(\vdash\mathbf{E}),$$

in which elimination happens after construction. Then the following partial derivation tree can alos be used to derive m_2:

$$\dfrac{\mathcal{C} \vdash \mathsf{pk}(\mathsf{m}_2) \quad \mathcal{C} \vdash \mathsf{n}_2}{\mathcal{C} \vdash \mathsf{m}_2}(\vdash\mathbf{E})$$

Although EC certainly holds if the above verification succeeds, the converse is probably not true. However, because the above verification is enough for situations that seem to occur in practice, we do not consider generalisations here.

Concerning the EE assumption, we note that it certainly holds for function symbols f/k that have just one construction rule

$$\mathsf{f}(x_1, \ldots, x_k) \leftarrow x_1, \ldots, x_k.$$

Namely, in this case, if the derivations of $\mathsf{m}_1, \mathsf{m}_2$ both end with a construction step, then the conclusion that $\mathsf{p}_1 \doteq \mathsf{p}_2$ can also be reached without constructing m_1 and m_2 by simply using their respective submessages that $\mathsf{p}_1, \mathsf{p}_2$ occur in.

However, if there are function symbols with multiple construction rules, then the EE assumption may no longer hold. In this case, the assumption needs to be verified by hand by showing that constructing the same message contents in two different ways does not lead

Algorithm 1 Given $\mathcal{C}$, compute view $(\mathsf{O}, \leftrightarrow)$ corresponding to $\mathcal{C}$

{find all m up to content equivalence such that $\mathcal{C} \vdash^{-} \mathsf{m}$}
$\mathcal{C}^{cnt} := \mathcal{C}$
repeat
  $\mathcal{C}^{old} := \mathcal{C}^{cnt}$
  for all $\mathsf{m} \in \mathcal{C}^{old}$, non-testing rules $m \xrightarrow{\doteq n_1, \ldots, \doteq n_l} n$, subst. σ s.t. $m\sigma = \mathsf{m}$ **do**
    if $n\sigma \notin \mathcal{C}^{cnt}$ and all $n_i\sigma$ can be constructed from $\mathcal{C}^{old}$ **then**
      $\mathcal{C}^{cnt} = \mathcal{C}^{cnt} \cup \{n\sigma\}$
    end if
  end for
until $\mathcal{C}^{cnt} = \mathcal{C}^{old}$
{find all m such that $\mathcal{C} \vdash^{-} \mathsf{m}$}
$\mathcal{C}^{el}$:=msg obtained from $\mathcal{C}$ by elimination {use $\mathcal{C}^{cnt}$ to construct aux. msg}

{compute detectable items, associable contexts}
O := (($\mathcal{C}^{el} \cap \mathsf{P}^{\mathrm{ctx}}$)+closure under equatability)$\cap \mathsf{O}^{\mathrm{ctx}}$
$\leftrightarrow$:= $\{*|^{\pi}_{k} \leftrightarrow *|^{\eta}_{l} \mid$ equatable context ids exist$\}$+sym./refl./trans. closure
return $(\mathsf{O}, \leftrightarrow)$

to new equatability conclusions. For instance, consider a symbol $\mathsf{dh}/2$[21] with construction rules

$$\mathsf{dh}(x,y) \leftarrow \mathsf{pk}(x), y \quad \mathsf{dh}(x,y) \leftarrow x, \mathsf{pk}(y).$$

[21] Intuitively, this models something similar to Diffie-Hellman key exchange. However, in contrast to real Diffie-Hellman, in this model, $\mathsf{dh}(x,y) \neq \mathsf{dh}(y,x)$ in general.

Suppose we have two differently-constructed instances of dh:

$$\frac{\mathcal{C} \vdash \mathsf{pk}(\mathsf{m}_1) \quad \mathcal{C} \vdash \mathsf{m}_2}{\mathcal{C} \vdash \mathsf{dh}(\mathsf{m}_1, \mathsf{m}_2)} (\vdash\mathbf{C}) \quad \frac{\mathcal{C} \vdash \mathsf{m}'_1 \quad \mathcal{C} \vdash \mathsf{pk}(\mathsf{m}'_2)}{\mathcal{C} \vdash \mathsf{dh}(\mathsf{m}'_1, \mathsf{m}'_2)} (\vdash\mathbf{C}).$$

To equate $\mathsf{m}_1@z$ to $\mathsf{m}'_1@z$, we can use $\mathsf{pk}(\mathsf{m}_1)$ on the left and construct it from m'_1 on the right; and similarly for m_2. Hence the EE assumption holds for this primitive.

3.7 *Computing Actor Views*

In this section, we discuss how to automatically compute actor views. Specifically, we give an algorithm that, given a knowledge base and a signature with associated construction and elimination rules, computes the actor view corresponding to that knowledge base and signature. To do this, it uses the alternative deductive system presented in the previous section. Algorithm 1 shows our algorithm. Based on this algorithm, we have developed a Prolog-based tool for formal analysis of privacy in communication protocols[22]. We briefly discuss the performance of the implementation in that tool.

[22] Available at `http://code.google.com/p/objective-privacy/`

The most complex task in our algorithm is to compute, given knowledge base $\mathcal{C}$, the set $\mathcal{C}^{el}$ of all messages m for which $\mathcal{C} \vdash^{-} \mathsf{m}$. Our basic idea is to start with $\mathcal{C}$, and iteratively add messages that can be obtained from messages we have by applying elimination rules. To determine the applicability of an elimination rule, we check whether its auxiliary messages can be obtained using construction

rules from messages we have derived so far. We improve on this basic idea by performing a small optimisation. Namely, we first iteratively find all messages obtainable by elimination rules as above, obtaining set $\mathcal{C}^{cnt}$; but we only make sure that the set is complete *up to content equivalence*. When this process is finished, we once again go through $\mathcal{C}$ to see which messages can be obtained by applying elimination rules; now using set $\mathcal{C}^{cnt}$ to see which auxiliary messages can be derived. This gives us the set $\mathcal{C}^{el}$ of all messages m such that $\mathcal{C} \vdash^{-} \mathsf{m}$. This optimisation makes the algorithm more efficient because, in the first (iterative) step, we do not need to consider testing rules, because they do not give us new message contents. Because of visible failure and reconstruction, there are typically many such rules.

Assuming EE and EC (see previous section), it easy to compute the detectable items O and associability relation $\leftrightarrow$ from $\mathcal{C}^{el}$. To find all detectable items, we consider all context items $\mathsf{p} \in \mathcal{C}^{el}$, and iteratively find all context items that they are equatable to until we reach a fixed point. This is done by inspecting all messages $\mathsf{m} \in \mathcal{C}^{el}$ and all locations z such that $\mathsf{m@}z = \mathsf{p}$; and looking for $\mathsf{m}' \in \mathcal{C}^{el}$ such that $\mathsf{m}' \doteq \mathsf{m}$: then $\mathsf{m}'@z$ is equatable to p. Similarly, to find all associable contexts, we start with an arbitrary context $*|_k^{\pi}$; for each identifier i occurring in the context, we find other contexts with equatable identifiers; and continue until we reach a fixed point.

It is clear that the above algorithm, if it terminates, computes the correct actor view corresponding to a knowledge base. To ensure termination, we need to assume that any message can be constructed in only finitely many ways, and only finitely many messages can be obtained from it using elimination rules. This ensures that if we start with a finite knowledge base and then try to iteratively find all messages that can be derived from it, we will inevitably reach a fixed point. Let $\rightarrow^*$ be the least transitive relation such that, if $\mathsf{m} \dashrightarrow \mathsf{n}$, then $\mathsf{m} \rightarrow^* \mathsf{n}$. Let $\leftarrow^*$ be the least transitive relation such that, if $\mathsf{m} \leftarrow \mathsf{n}_1, \ldots, \mathsf{n}_k$, then $\mathsf{m} \leftarrow^* \mathsf{n}_i$ for any i. We formalise the above assumption as follows:

Assumption 3.7.1 (Finiteness)**.** For any context message m, the sets $\{\mathsf{n} \mid \mathsf{m} \rightarrow^* \mathsf{n}\}$ and $\{\mathsf{n} \mid \mathsf{m} \leftarrow^* \mathsf{n}\}$ are finite.

We claim that:

Proposition 3.7.2. Let $\mathcal{C}$ be a knowledge base. Suppose the EC and EE assumptions are satisfied. If Algorithm 1 terminates, then it correctly computes the view corresponding to $\mathcal{C}$. Moreover, if the finiteness assumption is satisfied, then Algorithm 1 always terminates.

It is usually easy to see that the finiteness assumption is satisfied. For instance, for the standard primitives presented in Section 3.4, all elimination rules result in strict submessages of the original message, or in $\mathsf{pk}(x)$ where x is a strict submessage. This means that the set $\{\mathsf{n} \mid \mathsf{m} \rightarrow^* \mathsf{n}\}$ is a subset of the set of strict submessages and

pk's of them, hence finite. Similarly, for any set of primitives using just the standard construction rule $\mathsf{f}(a_1,\ldots,a_k)\leftarrow a_1,\ldots,a_k$, all messages to construct a message from are strict submessages, hence $\{\mathsf{n} \mid \mathsf{m} \leftarrow^* \mathsf{n}\}$ is a subset of[23] the set of strict submessages of m, which is finite.

We finish the discussion of our algorithm by giving an impression of the performance of our Prolog implementation. Namely, we took the knowledge bases of the actors in the four identity management architectures from Chapter 7, and timed how long it takes to compute the view from them. In Figure 3.5, we plot this time[24] against the "complexity" of the knowledge base, computed as the number of function symbols and context items occurring in it. As expected, the time taken increases considerably with the complexity of the knowledge base; but stays within reasonable bounds.

[23] (and in fact, equal to)

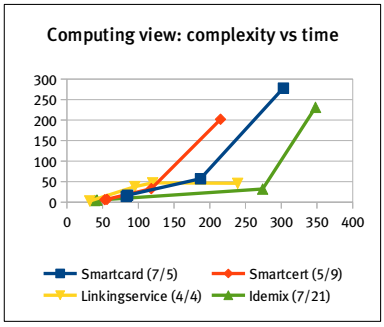


Figure 3.5: Performance numbers of our implementation for computing views: each point represents that the computation of a view from a knowledge base with given complexity (x-axis) took a given amount of time (y-axis, in ms). For each knowledge base, (x/y) indicates the number of primitives (x) and inference rules (y) used (lists and list elimination both counted only once)

[24] Experiments were run on the PC of the author: an Intel Core 2 Quad Q9400 at 2.66 GHz; only one processor was used

3.8 Discussion

Validity of our Model

The validity of the conclusions given by our reasoning model depends on two questions: does the reasoning model accurately model knowledge of personal information, and do the elimination rules accurately model cryptographic primitives.

Reasoning Model Concerning our model for reasoning about messages, the results match our intuitive expectations for the examples we have looked at; moreover, the relative simplicity of our reasoning model (only three rules) suggests that the modelled concepts are generic. Some additional confidence in its results can be obtained from comparisons with other models.

Our model is an adaptation of existing deductive systems from traditional contents-layer models to our three-layer model. At least, because the auxiliary messages for elimination rules are used up to contents, it is clear that our model succeeds in deriving *some* representation of any message whose contents could traditionally be derived. However, it remains a question if it can derive *all relevant* representations. Another frequently-used model for reasoning about knowledge[25] uses equational theories. In Chapter 4, we are able to show very precise correspondence results proving that our definitions can be interpreted in terms of well-known low-level definitions of knowledge in this setting.

[25] Abadi and Fournet (2001)

Our high-level definitions of associability and linkability are conceptually similar to existing "equivalence-based" privacy properties[26] defined using the applied pi calculus[27]. Conceptually, these properties also express links between protocol instances that can be made due to the use of identifiers. Technically, there are several differences. Primitives for equivalence-based privacy properties are typically modelled using equational theories instead of deduction rules. Also, properties are defined in terms of "equivalences" that

[26] E.g., Arapinis et al. (2010, 2012), Delaune et al. (2009), Dong et al. (2012, 2013), Dreier et al. (2013)

[27] Abadi and Fournet (2001); and Blanchet et al. (2008)

compare different scenarios and quantified over *all possible* evolutions of these scenarios in the presence of active attackers. Despite these differences, there is still an intuitive correspondence between equivalence-based privacy properties and knowledge about personal information in our framework. We discuss this correspondence in Section 9.3.

Model of Primitives To obtain an accurate model of a cryptographic primitive, existing formalisations using deductive systems can be used to some degree. Since deductive systems are traditionally used at the contents layer, existing formalisations of primitives do not consider testing rules (e.g., for signature verification), because such rules have no contents-layer effect. Moreover, such formalisations typically do not distinguish between deterministic and probabilistic primitives because this difference is often irrelevant for the type of property considered[28]. However, both aspects are relevant for us, so to obtain an accurate model from an existing deductive model, it will usually be needed to inspect the underlying cryptography.

[28] Cortier et al. (2007)

Existing formalisations using equational theories may be more useful for adaptation because they do consider the above aspects. However, because equational theories are more powerful than the model presented above, adaptation may not always be possible, or only possible by considering simplifications. We discuss this in Chapter 4.

Finally, in case no suitable formalisation exists already, it can be developed based on the cryptographic details of the primitive. This is not easy and, given the simplicity of symbolic models compared to the original cryptography, it inevitably involves making simplifying assumptions. However, for many primitives it should be possible to come up with a model that safely approximates privacy aspects of the primitive. As an example, we refer to our models of zero-knowledge proofs and anonymous credentials (§6.4–§6.5).

Our model of standard cryptographic primitives presented in Section 3.4 is similar to many existing models. We postpone the technical justification to Chapter 4, but mention now that our models are equivalent to the equational models suggested for use with the ProVerif tool[29]. In other work[30], equational models of several cryptographic primitives are presented that are justified with respect to the lower-level "computational model" of cryptography. Our models of symmetric and asymmetric encryption are equivalent to those equational models, while our model of concatenation is a safe approximation.

[29] Blanchet and Smyth (2011)

[30] Baudet et al. (2010)

Technical Limitations

Our formalism has several technical limitations on what kind of cryptographic primitives and operations can be modelled. We now discuss these limitations in detail.

Construction and Elimination Rules We model cryptography using construction and elimination rules; due to our uniqueness assumptions, no two messages can have the same contents apart from those built using construction rules for the same primitive. The combination of having only rules of this type and imposing uniqueness of messages makes certain aspects of cryptography impossible to model.

First, we cannot model that messages represented by different terms have the same contents. This means that, for instance, we cannot model an "exclusive or" primitive $\oplus$ because this primitive has the property that the contents of $(a \oplus b) \oplus b$ and a coincide. This limitation is probably not easy to overcome in general; indeed, as noted in a survey on such "algebraic properties" of cryptographic primitives[31], not many tools for protocol verification can handle them.

[31] Cortier et al. (2006)

Also, our definitions of construction and elimination rules (Section 3.3) only allow rules of a certain form. In particular, in a construction rule, variables may occur only once both in the result and in its parts; in elimination rules, variables may occur only once in whole messages. Not imposing such restrictions would open the door to elimination rules such as $\mathsf{f}(a,a) \rightarrow a$ of which it is not clear how the instantiation should be defined. Because the cryptographic operations we model by these rules act on message contents, the most obvious interpretation would be that $\mathsf{f}(\mathsf{m}_1, \mathsf{m}_2)$ instantiates $\mathsf{f}(a,a)$ whenever $\mathsf{m}_1 \doteq \mathsf{m}_2$. Then, both m_1 and m_2 are valid results of applying the cryptographic operation modelled by this elimination rule. Also, the fact that the rule applies should tell the actor that $\mathcal{C} \vdash \mathsf{m}_1 \doteq \mathsf{m}_2$. This latter conclusion means that implementing this interpretation would require more advanced content analysis reasoning. For simplicity, we do not consider this possibility.

Certain other knowledge about cryptographic primitives cannot be expressed at all in terms of constructing or destructing messages. For instance, Corin et al.[32] show how to model encryption schemes that are "which-key revealing", i.e., that allow an actor to see whether two given ciphertexts have been encrypted using the same key. Such properties do not fit into the construction/elimination framework, and hence cannot be captured in our model.

[32] Corin et al. (2005)

Finally, we remark that the use of multiple construction rules to capture different ways in which a cryptographic primitive can be built is not common in the literature. For instance, in equational theory models, function symbols f/k always have just one "construction rule" $\mathsf{f}(x_1, \ldots, x_k) \leftarrow x_1, \ldots, x_k$; alternative constructions are modelled by equations. In the case of our model, it is frequently possible to convert a signature that uses multiple construction rules into an equivalent[33] signature that does not. However, we note that this would generally break the EC and EE assumptions that the implementation of our reasoning system relies on. In particular, suppose we have a signature in which function symbols always have the

[33] In the sense that the resulting actor views are the same

standard construction rule $\mathsf{f}(x_1,\ldots,x_k)\leftarrow x_1,\ldots,x_k$ and possibly additional construction rules $\mathsf{f}(a_1,\ldots,a_k)\leftarrow b_1,\ldots,b_l$. In this case, we can construct an equivalent signature with just standard construction rules by replacing every non-standard constructor rule $\mathsf{f}(a_1,\ldots,a_k)\leftarrow b_1,\ldots,b_l$ by

- function symbol f_i with constructor rule $\mathsf{f}_i(x_1,\ldots,x_l)\leftarrow x_1,\ldots,x_l$, where x_i are distinct variables;
- elimination rule $\mathsf{f}_i(b_1,\ldots,b_l)\rightarrow\mathsf{f}(a_1,\ldots,a_k)$.

(Reconstruction rules $\mathsf{f}(a_1,\ldots,a_k)\xrightarrow{\doteq b_1,\ldots,\doteq b_l}b_i$ stay intact.) As long as only function symbol f is used in knowledge bases, it can be shown that this model is equivalent to the original one. However, because it may be needed to construct f_i before eliminating to f, this alternative model in general breaks the EC and EE assumptions, in which case it cannot be used with our implementation.

Visible Failure Our visible failure assumption states that an actor can see if a cryptographic operation has succeeded or not. For many cryptographic primitives and operations, this is a reasonable assumption: for instance, for asymmetric encryption and digital signatures as presented in this chapter. Symmetric encryption exists both in "authenticated" and "non-authenticated" variants[34]: the first variant satisfies visible failure whereas the second one does not. As remarked in Section 3.3, even if the primitive itself does not satisfy visible failure, actors are often able to tell a correct from a wrong result, hence visible failure is in practice only a slight overapproximation of actor's knowledge.

[34] Bellare and Namprempre (2008)

Nonetheless, it should be possible to adapt our model to the non-visible-failure situation by slightly changing the elimination rules that the assumption introduces. For instance, for encryption, we could use the following rule to capture the observation that a certain key was used:

$$\mathsf{enc}(m,k)\xrightarrow{\doteq m,\doteq k}k;$$

namely, the key can be learned only if the resulting message is recognised (i.e., can be constructed). Note that, in order for this to work, the contents of all messages that an actor can recognize without having seen them before, should be explicitly added to his knowledge base.

EC, EE assumptions In order to actually compute actor views, we needed the EC and EE assumptions. Although both hold for the primitives we have come across in practice, they certainly do not hold in general.

The EC assumption states that it is never necessary to apply an elimination rule to a constructed message. The following example shows a model of a primitive in which this assumption does not hold:

Example 3.8.1. Consider signature $\Sigma = \{\mathsf{tw}/1\}$ with elimination rule

$$\mathsf{tw}(\mathsf{tw}(x)) \rightarrow x,$$

i.e., if tw is applied twice to a message m, the original message can be recovered from the result. This rule does not satisfy EC: the only way to obtain p from $\mathsf{tw}(\mathsf{p})$ is by first constructing $\mathsf{tw}(\mathsf{tw}(\mathsf{p}))$ and then applying the elimination rule. Indeed, we have $\{\mathsf{tw}(\mathsf{p})\} \vdash \mathsf{p}$ but not $\{\mathsf{tw}(\mathsf{p})\} \vdash^{-} \mathsf{p}$. □

In the above example, note that signature $\Sigma = \{\mathsf{tw}/1\}$ with elimination rules

$$\mathsf{tw}(\mathsf{tw}(x)) \rightarrow x, \quad \mathsf{tw}(x) \rightarrow x$$

has an equivalent derivability relation, but does satisfy EC. It is interesting if such a "saturation" procedure is possible in general.

The EE assumption states that it is not possible to equate additional pieces of information by constructing a message in two different ways. The primitive in the following example breaks that assumption:

Example 3.8.2. Consider signature $\Sigma = \{\mathsf{x}/2, \mathsf{i}/1, \mathsf{j}/2\}$ with construction rules

$$\mathsf{x}(a,b) \leftarrow a, \mathsf{i}(b) \quad \mathsf{x}(a,b) \leftarrow a, \mathsf{j}(b)$$

and elimination rules

$$\mathsf{x}(a,b) \xrightarrow{\doteq a, \doteq \mathsf{i}(b)} a \quad \mathsf{x}(a,b) \xrightarrow{\doteq a, \doteq \mathsf{i}(b)} \mathsf{i}(b)$$

$$\mathsf{x}(a,b) \xrightarrow{\doteq a, \doteq \mathsf{j}(b)} a \quad \mathsf{x}(a,b) \xrightarrow{\doteq a, \doteq \mathsf{j}(b)} \mathsf{j}(b)$$

due to reconstruction. Clearly, if $\mathsf{p} \doteq \mathsf{p}'$, then

$$\{\mathsf{j}(\mathsf{p}), \mathsf{i}(\mathsf{p}')\} \vdash \mathsf{p} \doteq \mathsf{p}',$$

e.g., by constructing $\mathsf{x}(\mathsf{j}(\mathsf{p}), \mathsf{p})$ and $\mathsf{x}(\mathsf{j}(\mathsf{p}), \mathsf{p}')$. This conclusion clearly cannot be reached without constructing some $\mathsf{x}(\cdot,\cdot)$, breaking EE. □

Evolution of the Model

The reasoning model presented in this section has evolved considerably since we first published about it[35]. Originally, the deductive system was instantiated for a particular set of primitives, and equatability (Definition 3.5.2) was included in the deductive system rather than built on top of it. Also, the interplay between testing and elimination rules was different: auxiliary messages for elimination rules were not defined up to contents, but would instead need to be obtained by applying a testing rule first[36]. However, despite these differences, the resulting view of an actor has remained the same. Intuitively, application of an elimination rule in our present deductive system corresponds to application of an elimination rule with possible testing rules applied first in the original deductive system. Similarly, any application of the content analysis rule in the original deductive system can be moved to the end of the deduction[37], hence equivalently be defined on top of the deductive system rather than included in it.

[35] Veeningen et al. (2011b)

[36] Veeningen et al. (2011b); and Veeningen et al. (2014)

[37] Veeningen et al. (2014)

4
Detectability and Linkability with Equational Theories

Contents

ALTHOUGH THE PREVIOUS CHAPTER presents a reasoning system to determine knowledge from messages, this system is limited in what kinds of cryptographic primitives and operations can be modelled. In particular, as noted in Section 3.8, the uniqueness assumption means that we cannot model cryptographic primitives that satisfy certain "algebraic properties". For instance, we cannot model an "exclusive or" primitive $\oplus$ because this primitive has the property that the contents of $(a \oplus b) \oplus b$ and a coincide. Also, the visible failure assumption means that we can model primitives like non-authenticated[1] encryption only by over-approximating the knowledge of actors. Finally, by restricting ourselves to construction and elimination rules, we cannot model possible checks that relate different messages to each other. For instance, we cannot model encryption schemes that are "which-key revealing"[2], i.e., that allow an actor to see whether two given ciphertexts have been encrypted using the same key.

[1] Bellare and Namprempre (2008)

[2] Corin et al. (2005)

An alternative way of modelling cryptographic primitives is by using an equational theory[3]. As in our model, messages using cryptographic primitives are modelled as terms; however, in this case, cryptographic operations are modelled as equations on these terms. For instance, we can use $\mathsf{enc}(m, k)$ to represent the encryption of message m using key k; the equation $\mathsf{dec}(\mathsf{enc}(m, k), k) = m$ represents that the decryption of $\mathsf{enc}(m, k)$ using key k returns m. This more general model at least partially solves the limitations above. For instance, the coincidence of $(a \oplus b) \oplus b$ and a can be modelled by the equation $\mathsf{xor}(\mathsf{xor}(a, b), b) = a$. Many cryptographic primitives have

[3] Abadi and Fournet (2001)

been modelled using equations: the standard ones from the previous chapter, but also more complicated ones like commitments, blind signatures[4], and non-interactive zero-knowledge proofs[5]. Equational theories have been used in various tools to verify properties of cryptographic protocols, e.g., ProVerif[6] and KiSS[7].

In this chapter, we show how knowledge of personal information from messages can alternatively be derived using equational theories. We present alternative definitions for detectability and linkability based on equations instead of inference rules. We do this by noting similarities between our notion of equatability and the existing notion of resistance to guessing attacks[8]. This way, detectability and linkability are also defined in cases beyond the limitations set out above. We also show that, when primitives *do* fall within these limitations, the equation-based and inference rule-based definitions coincide. Thus, the equation-based definitions are suitable generalisations of our previous framework. Finally, we present our tool that automatically translates detectability and linkability queries to standard queries about knowledge with equational theories, and interfaces the KiSS[9] tool to find the answer. Hence, we obtain an alternative way to automatically determine knowledge of personal information based on communication; although it is more general, this will turn out to come at a price in terms of practical performance.

[4] Both primitives: Delaune et al. (2009)

[5] Backes et al. (2008)

[6] Blanchet et al. (2008)

[7] Ciobâcă et al. (2009)

[8] Corin et al. (2005); and Delaune et al. (2008)

[9] Ciobâcă et al. (2009)

4

Outline In this chapter:

- We show how knowledge about contents of cryptographic messages is defined using the standard concepts of deducibility and static equivalence based on equational theories (§4.1);
- We show how resistance to guessing attacks can be defined both in the equational model of this chapter and in the rule-based model of Chapter 3, and draw parallels between the two models (§4.2);
- We use these parallels to define the view corresponding to knowledge of messages in the equational model (§4.3);
- We show how a rule-based model of cryptographic primitives can be converted to an equational model (§4.4), and prove that, in this case, the views obtained using the rule-based and equational models coincide (§4.5);
- We briefly discuss our implementation for computing actor views in the equational model (§4.6);
- Finally, we discuss what equational theories are suitable for privacy analysis, and elaborate on the relation between the rule-based and equational models (§4.7).

4.1 Actor Knowledge with Equational Theories

In this section, we introduce standard notions[10] for defining actor knowledge using equational theories. Let $\mathcal{N} = \{a, b, c, \ldots\}$ be an infinite set of *names*, representing pieces of information like keys, nonces, data items, or identifiers. Names should be thought of as contents of a particular piece of information; thus, intuitively, names correspond to content-layer items in our three-layer model. Similarly to our model, cryptographic primitives are modelled by an *equational signature* Σ^{eq} consisting of function symbols f with associated arity[11] $k \geq 0$, denoted f/k. The set $\mathcal{T}_{\mathcal{N}}$ of *ground terms* representing cryptographic messages is built from the set $\mathcal{N}$ of names by recursive application of function symbols. In contrast to our model, cryptographic operations are not modelled as rules, but as function symbols. For instance, the function symbol $\mathsf{dec}/2$ can be used to represent the decryption operation: then $\mathsf{dec}(x, k)$ represents the decryption of message x with key k (regardless of whether x is actually an encryption with key k).

The following example defines an equational signature modelling some basic cryptographic primitives and operations, and shows some examples of ground terms:

Example 4.1.1. The equational signature

$$\Sigma^{\text{eq}} = \{\mathsf{pair}/2, \mathsf{enc}/2, \mathsf{penc}/3, \mathsf{pk}/1, \mathsf{fst}/1, \mathsf{snd}/1, \mathsf{dec}/2, \mathsf{pdec}/2\}$$

models concatenation, deterministic symmetric encryption, nondeterministic asymmetric encryption, public keys, and their respective cryptographic operations.

To model concatenation, we use three function symbols: $\mathsf{pair}/2$ to model the concatenation of two messages; $\mathsf{fst}/1$ to model extraction of the first element of a pair; and $\mathsf{snd}/1$ to model extraction of the second element. For instance, let a, b be names; then the ground term $\mathsf{pair}(\mathsf{pair}(a, b), a)$ represents the message obtained by first concatenating a and b, and then concatenating the result and a; $\mathsf{fst}(\mathsf{pair}(a, b))$ represents the message obtained by extracting the first element from concatenation $\mathsf{pair}(a, b)$ of a and b.

Deterministic symmetric encryption is modelled by function symbols $\mathsf{enc}/2$ and $\mathsf{dec}/2$, where $\mathsf{enc}(m, k)$ represents the encryption of message m under key k, and $\mathsf{dec}(m, k)$ represents the decryption of message m under key k.

Probabilistic asymmetric encryption is modelled by function symbols $\mathsf{penc}/3$, $\mathsf{pdec}/2$, and $\mathsf{pk}/1$: $\mathsf{pk}(x)$ represents the public key corresponding to private key x; $\mathsf{penc}(m, pk, r)$ represents the encryption of message m using public key pk and randomness r; $\mathsf{pdec}(m, sk)$ represents the decryption of message m using private key sk.[12]

Cryptographic primitives can be arbitrarily combined. For instance, let m and k be names. Then $\mathsf{pair}(\mathsf{dec}(\mathsf{enc}(m, k), k), m)$ represents the message obtained by encrypting m by k; decrypting the result with k; and concatenating the result with m. □

[10] E.g, see Delaune et al. (2008)

[11] I.e., number of parameters. Symbols with arity 0 typically do not occur in messages but are used to model the result of "tests": cf. Example 4.1.8 and Hüttel and Pedersen (2007).

[12] Alternatively, probabilistic asymmetric encryption can be modelled by first modelling deterministic asymmetric encryption and then explicitly adding randomness like in Section 3.4; the model presented here is chosen because it will help to illustrate some technical points later.

The functionality of cryptographic primitives is modelled by an equational theory. Formally, let $\mathcal{X}$ be an infinite set of *variable*; and let $\mathcal{T}_{\mathcal{X}}$ denote the set of *variable terms*: terms built from variables in $\mathcal{X}$ using the function symbols in equational signature Σ^{eq}. An *equational theory* E is a finite set of equations $U = V$, where $U, V \in \mathcal{T}_{\mathcal{X}}$ are variable terms. We define $=_{\mathsf{E}}$ to be the least equivalence relation on ground terms such that:

- For every substitution σ of ground terms for variables in U, V[13] such that $U\sigma$ and $V\sigma$ are ground terms: $U\sigma =_{\mathsf{E}} V\sigma$;
- If $t_1 =_{\mathsf{E}} t'_1, \ldots, t_k =_{\mathsf{E}} t'_k$, then $\mathsf{f}(t_1, \ldots, t_k) =_{\mathsf{E}} \mathsf{f}(t'_1, \ldots, t'_k)$.

[13] I.e., a set $\sigma = \{u_1 \to t_1, \ldots, u_k \to t_k\}$ where $u_i \in \mathcal{X}$, $t_i \in \mathcal{T}_{\mathcal{N}}$. Substitution σ acts on $X \in \mathcal{T}_{\mathcal{X}}$ by replacing all variables by corresponding terms. The result is denoted denoted $X\sigma$.

We write $\neq_{\mathsf{E}}$ for its complement. Intuitively, given two ground terms t_1 and t_2, $t_1 =_{\mathsf{E}} t_2$ captures that the two terms represent messages with the same contents.

The following example demonstrates how the functionality of cryptographic primitives is captured by an equational theory:

Example 4.1.2. Consider the equational signature Σ^{eq} from Example 4.1.1:

$$\Sigma^{\text{eq}} = \{\mathsf{pair}/2, \mathsf{enc}/2, \mathsf{pk}/1, \mathsf{penc}/3, \mathsf{fst}/1, \mathsf{snd}/1, \mathsf{dec}/2, \mathsf{pdec}/2\}$$

We now define a corresponding equational theory E.

Let x, y, z denote variables. The extraction of the first and second element of a pair are modelled by equations $\mathsf{fst}(\mathsf{pair}(x, y)) = x$ and $\mathsf{snd}(\mathsf{pair}(x, y)) = y$, respectively. For instance, $\mathsf{fst}(\mathsf{pair}(a, b)) =_{\mathsf{E}} a$ by substituting a for x and b for y, respectively.

The decryption operation for deterministic symmetric encryption is modelled by equation $\mathsf{dec}(\mathsf{enc}(x, y), y) = x$: that is, decryption of any message $\mathsf{enc}(x, y)$ using key y gives plaintext x. For instance, $\mathsf{pair}(\mathsf{dec}(\mathsf{enc}(m, k), k), m) =_{\mathsf{E}} \mathsf{pair}(m, m)$ because $\mathsf{dec}(\mathsf{enc}(m, k), k) =_{\mathsf{E}} m$ and $m =_{\mathsf{E}} m$. Note that $\mathsf{pair}(\mathsf{dec}(\mathsf{enc}(m, k), l), m)$ for $l \neq k$ is also a valid ground term: it represents the result when the decryption operation is applied using a wrong key. Unlike in the previous case, it is not equivalent under $=_{\mathsf{E}}$ to a "simpler" message.

The decryption operation for non-deterministic asymmetric encryption is modelled by equation $\mathsf{pdec}(\mathsf{penc}(x, \mathsf{pk}(y), z), y) = x$; that is, an encryption under a public key can be decrypted using the corresponding private key. □

Unless otherwise indicated, we will use the equational signature Σ^{eq} and equational theory E from the above example for the examples in the remainder of this section.

A *frame* captures the knowledge of an actor at a certain point in time. It is a structure

$$\phi = \nu n_1, \ldots, n_l.\{m_1/x_1, \ldots, m_k/x_k\}.$$

Here $\{n_1, \ldots, n_l\} \subset \mathcal{N}$ is a set of *restricted* names: names that the actor does not know a priori (e.g., other actors' nonces and private keys). If $l = 0$, we omit "ν.". Ground terms $m_i \in \mathcal{T}_{\mathcal{N}}$ correspond to messages

the actor has observed; variables $x_i \in \mathcal{X}$ are used to refer to these messages. (By assigning a variable to each message instance, different instances of the same message received at different moments in time can be distinguished.)

The next example demonstrates the notion of frame:

Example 4.1.3. Consider frames

$$\phi_1 = \nu n.\{\mathsf{enc}(n,k)/x_1\};\ \ \phi_2 = \nu n.\{n/x_1\};\ \ \phi_3 = \{\mathsf{enc}(m,k)/x_1\}.$$

The first frame represents the knowledge of an actor consisting of an encryption of an unknown value n under a known key k. The second frame represents the knowledge of a single, a priori unknown, value n. The third frame represents the knowledge of a known value m under a known key k. □

We now define the concept of *deducibility*[14], capturing if a message can be determined from a frame. Let

$$\phi = \nu n_1, \ldots, n_l.\{m_1/x_1, \ldots, m_k/x_k\}$$

be a frame. An actor who knows ϕ, knows all non-restricted names (i.e., all names except $n_1, \ldots, n_l$), and all messages corresponding to variables $x_1, \ldots, x_k$. A *recipe* $N \in \mathcal{T}_{\mathcal{X}\cup\mathcal{N}}$ for ϕ is a term built from non-restricted names and variables $x_1, \ldots, x_k$; intuitively, recipes represent all possible ways in which an actor can use his knowledge to build messages. Frame ϕ acts on recipe N as a substitution, i.e., by replacing all variables in N by their messages from ϕ; the result is denoted $N\phi$. Hence, the messages an actor can determine are exactly the results $N\phi$ he can obtain by applying recipes N to his frame ϕ. Deducibility is formally defined as follows:

[14] Delaune et al. (2008)

Definition 4.1.4. Let ϕ be a frame, and $M \in \mathcal{T}_{\mathcal{N}}$ a ground term. We say that M is *deducible* from ϕ, denoted $\phi \vdash M$, if there exists a recipe $N \in \mathcal{T}_{\mathcal{X}\cup\mathcal{N}}$ for ϕ such that $N\phi =_{\mathsf{E}} M$.

The next example demonstrates deducibility:

Example 4.1.5. Consider again the frames from Example 4.1.3:

$$\phi_1 = \nu n.\{\mathsf{enc}(n,k)/x_1\};\ \ \phi_2 = \nu n.\{n/x_1\};\ \ \phi_3 = \{\mathsf{enc}(m,k)/x_1\}.$$

In frame ϕ_1, name k is not restricted, so the actor can obtain the value of n by decrypting known message x_1. Formally, $\phi_1 \vdash n$ holds using recipe $N = \mathsf{dec}(x_1,k)$ because $N\phi = \mathsf{dec}(\mathsf{enc}(n,k),k) =_{\mathsf{E}} n$. In frame ϕ_2, n occurs as a message by itself, so $\phi_2 \vdash n$ using recipe x_1. In frame ϕ_3, n is not restricted, so it is assumed to be known by the actor: $\phi_3 \vdash n$ using recipe n.[15] □

[15] In fact, it can be shown that the sets of messages deducible from ϕ_1, ϕ_2, ϕ_3 are all the same.

Deducibility is a standard notion from the literature that can be used, for instance, to define secrecy properties[16] of cryptographic protocols: intuitively, secrecy of a piece of information k in a system is satisfied if, in all frames arising from evolutions of the system, an attacker cannot deduce k.

[16] Abadi and Blanchet (2005)

A second concept capturing knowledge in the equational setting is *static equivalence*. Intuitively, two frames are called statically equivalent if an actor cannot see the difference between them. That is, suppose the actor is asked if he is in the situation modelled by frame ϕ_1 or in the situation modelled by ϕ_2. Clearly, for the situations to look the same, the frames need to have the same number of messages. Static equivalence additionally requires that, whenever the actor can derive some information from messages in ϕ_1 that he can recognise (e.g. a known name, or another message), then applying the same recipe on the respective messages in ϕ_2 should (informally) "give the same result", and vice versa. Hence, the actor cannot recognise anything that tells him if he is in situation ϕ_1 or ϕ_2.

Privacy properties are formalised as static equivalences by comparing two frames that coincide on publicly known information, but differ on information that should remain hidden. For instance, suppose an actor in an e-voting system is allowed to know the contents of individual votes and the identities of individual voters, but not who cast which vote. This can be modelled by static equivalence of a frame in which voter 1 has cast vote A and voter 2 has cast vote B to a frame in which voter 1 has cast vote B and voter 2 has cast vote A[17]. Indeed, these two frames coincide on public information (the identities of the voters and the contents of the votes are the same in both frames), but differ on private information (which particular voter has cast which particular vote). Static equivalence means that the actor cannot distinguish the two situations, hence he cannot determine who cast which vote.

[17] Delaune et al. (2009)

Formally, two frames ϕ_1, ϕ_2 are statically equivalent if all comparisons an actor can make hold in ϕ_1 if and only if they hold in ϕ_2. We first define what we mean by a comparison. Namely, let ϕ be a frame, and let M and N be two recipes. If M, N are recipes for ϕ, i.e., they do not contain names that are restricted in ϕ, then M and N are *equal* in frame ϕ, denoted $(M = N)\phi$, if $M\phi =_{\mathsf{E}} N\phi$ (otherwise: $(M \neq N)\phi$). For instance, letting $\phi = \{k/x_1\}$, we have $(k = x_1)\phi$. Now, for ϕ_1 and ϕ_2 to be statically equivalent, we require that the same comparisons hold in both frames, i.e., $(M = N)\phi_1$ if and only if $(M = N)\phi_2$. Note that it may happen that M, N are valid recipes for ϕ_1, but not for ϕ_2 because they contain names that are restricted in ϕ_2 (or the other way round). To define $(M = N)\phi_2$ in this case, consider frame ϕ' obtained from ϕ by renaming these restricted names. Now, $(M = N)\phi_2$ if $M\phi' =_{\mathsf{E}} N\phi'$[18]. For instance, consider frame $\psi = \nu k.\{k/x_1\}$. Now, $(k = x_1)\psi$ does not hold: consider frame $\psi' = \nu l.\{l/x_1\}$, then $k\psi' =_{\mathsf{E}} x_1\psi'$ does not hold. Static equivalence is formally defined as follows:

[18] Note that this is well-defined because $=_{\mathsf{E}}$ does not depend on how we performed the renaming

Definition 4.1.6. Let ϕ and ψ be two frames

$$\phi = \nu n_1, \ldots, n_l.\{m_1/x_1, \ldots, m_k/x_k\},$$
$$\psi = \nu n'_1, \ldots, n'_m.\{m'_1/x_1, \ldots, m'_k/x_k\}$$

that have the same variables $x_1, \ldots, x_k$. Then ϕ and ψ are called

statically equivalent, denoted $\phi \approx \psi$, if for any recipes M, N,

$$(M = N)\phi \text{ iff } (M = N)\psi.$$

The next example demonstrates static equivalence.

Example 4.1.7. Consider again the frames from Example 4.1.3:

$$\phi_1 = \nu n.\{\mathsf{enc}(n,k)/x_1\};\ \phi_2 = \nu n.\{n/x_1\};\ \phi_3 = \{\mathsf{enc}(m,k)/x_1\}.$$

Intuitively, it is clear that ϕ_3 is not statically equivalent to ϕ_1 and ϕ_2. Namely, an actor can see the difference between an encryption under a known key of a known value (in ϕ_3) from a different value (in ϕ_2) or an encryption of a different value (in ϕ_1). In terms of comparisons between messages: he can use the known key k to encrypt the known value m, and compare it to the message x_1. In frame ϕ_3 this comparison will succeed; in the two other frames, it will not. Formally, take $M = x_1$ and $N = \mathsf{enc}(m,k)$: then $(M = N)\phi_3$, but $(M \neq N)\phi_1$ and $(M \neq N)\phi_2$. We conclude that $\phi_1 \not\approx \phi_3$ and $\phi_2 \not\approx \phi_3$.

Also, frames ϕ_1 and ϕ_2 are not statically equivalent to each other, meaning that equational theory E models an encryption scheme in which an actor can see the difference between an unknown value and the encryption of an unknown value using a known key. This is perhaps surprising; the formal reason is that, in this model, if we decrypt an encryption using the correct key, and then use the key again to re-encrypt the message, we get the original encryption. However, if we decrypt a random value using this key and re-encrypt, we do not get the original random value. Formally, take $M = x_1$ and $N = \mathsf{enc}(\mathsf{dec}(x_1,k),k)$. Now:

$$M\phi_1 = \mathsf{enc}(n,k) =_{\mathsf{E}} \mathsf{enc}(\mathsf{dec}(\mathsf{enc}(n,k),k),k) = N\phi_1;$$
$$M\phi_2 = n \neq_{\mathsf{E}} \mathsf{enc}(\mathsf{dec}(n,k),k) = N\phi_2.$$

Hence $\phi_1 \not\approx \phi_2$. □

In the previous chapter, we discussed "visible failure": i.e., the ability of an actor to see whether the correct inputs (e.g., the correct decryption key) were given to a cryptographic operation. The next example highlights how visible failure is modelled using equational theories.

Example 4.1.8. In Example 4.1.7, we showed that equational theory E models symmetric encryption in which an actor can see the difference between a random value and the an encryption of a random value with a known key. By the same line of reasoning, an actor can also see the difference between an encryption of a random value n with a known key k and an encryption of the same value with another key. That is, the decryption operation satisfies visible failure. Namely, we have that $\nu n.\{\mathsf{enc}(n,k)/x_1\} \not\approx \nu n.\{\mathsf{enc}(n,l)/x_1\}$ using recipes $M = \mathsf{enc}(\mathsf{dec}(x_1,k),k)$ and $N = x_1$.

In order to model decryption without visible failure, we should make sure that re-encrypting a decryption using the wrong key also

returns the original encryption. This can be achieved with equation $\mathsf{enc}(\mathsf{dec}(x,y),y) = x$. Indeed, given equational theory $\mathsf{E}' = \mathsf{E} \cup \{\mathsf{enc}(\mathsf{dec}(x,y),y) = x\}$, we have that

$$\nu n.\{\mathsf{enc}(n,k)/x_1\} \approx \nu n.\{\mathsf{enc}(n,l)/x_1\}.$$

Note that, necessarily, an actor can also no longer distinguish an encrypted random value from a random value, i.e., also frames ϕ_1 and ϕ_2 from the previous example are statically equivalent under this equational theory. (Note that decryption without visible failure cannot be expressed in the rule-based model of the previous chapter.)

On the other hand, the decryption operation for asymmetric encryption in E does not satisfy visible failure: because the actor cannot obtain the randomness, he cannot reconstruct the encryption. Indeed,

$$\nu n_1, n_2.\{\mathsf{penc}(n_1,\mathsf{pk}(k),n_2)/x_1\} \approx \nu n_1, n_2.\{\mathsf{penc}(n_1,\mathsf{pk}(l),n_2)/x_1\}.$$

To model asymmetric encryption with visible failure, one can add an explicit "test"[19]: two new function symbols $\mathsf{test\_pdec}/2, \mathsf{ok}/0$ and equation $\mathsf{test\_pdec}(\mathsf{penc}(x,\mathsf{pk}(y),z),y) = \mathsf{ok}$. Indeed, we have

[19] Hüttel and Pedersen (2007)

$$\nu n_1, n_2.\{\mathsf{penc}(n_1,\mathsf{pk}(k),n_2)/x_1\} \not\approx \nu n_1, n_2.\{\mathsf{penc}(n_1,\mathsf{pk}(l),n_2)/x_1\}$$

with respect to equational signature $\Sigma^{\mathrm{eq}\prime}$ and equational theory E'' obtained by including the above test. □

"Tests" similar to the one in the above example can also be used to model other assumptions on cryptographic primitives, like the "which-key revealing" property mentioned in the introduction to this chapter. See Corin et al.[20] for models of various assumptions on symmetric and asymmetric encryptions that can be modelled in this way.

[20] Corin et al. (2005)

4.2 *Resistance to Guessing Attacks*

In this section, we use resistance to guessing attacks to illustrate an intuitive correspondence between our rule-based model and the equational model above. We say that a protocol involving (human-chosen) passwords admits a guessing attack if an attacker who observes just one instance of the protocol, can use this to verify any number of guesses that he makes for the password. For instance, consider a protocol in which a password is sent with a cryptographic hash function applied to it. An attacker who intercepts this message, cannot directly deduce the password from it. However, he can make a guess, compute its cryptographic hash, and compare it to the hashed password. If the result is the same, then his guess was (very probably) correct. He can repeat this any number of times until he has found the correct password, hence this protocol indeed admits a guessing attack. On the other hand, a protocol in which the password is sent using a probabilistic encryption for which the attacker does not know the key and randomness does not allow such a guess.

In the equational setting, *resistance to guessing attacks*[21] models the non-existence of such attacks. For instance, consider the above example. We model the cryptographic hash function with function symbol $\mathsf{h}/1$ without equations. The intercepted message is modelled by frame $\phi = \nu pw.\{\mathsf{h}(pw)/x_1\}$, where pw represents the password. Note that the password cannot be deduced from this frame (i.e., $\phi \vdash pw$ does not hold), hence non-deducibility is not enough to protect against guessing attacks. However, we can formally model the guessing attack discussed above by noting that, for any guess x of the password, its correctness can be verified by checking if $\mathsf{h}(x) =_{\mathsf{E}} x_1$[22]. Based on this observation, resistance to guessing attacks can be modelled as a static equivalence property. Namely, we add the guess to the frame ϕ as a new variable $guess$, and then check for static equivalence between the case when the guess was correct (i.e., equal to pw) and the case when his guess was incorrect (i.e., equal to some new restricted name w representing a "wrong guess"):

[21] Corin et al. (2005); and Delaune et al. (2008)

[22] This is an idealised model of hash functions that does not consider collisions, hence any guess x satisfying $\mathsf{h}(x) =_{\mathsf{E}} x_1$ equals the password.

Definition 4.2.1. Let $\phi = \nu n_1, \ldots, n_k.\{m_1/x_1, \ldots, m_l/x_l\}$ be a frame, and n_i a restricted name of ϕ. We say that ϕ *is resistant to guessing attacks against* n_i if:[23]

$$\nu n_1, \ldots, n_k\{m_1/x_1, \ldots, m_l/x_l, n_i/guess\} \approx$$
$$\nu n_1, \ldots, n_k, w.\{m_1/x_1, \ldots, m_l/x_l, w/guess\}.$$

[23] Our definition is phrased slightly differently than the one in the literature (cf. Corin et al. (2005), Delaune et al. (2008)), but it is easily seen to be equivalent.

Note that, as discussed in the previous section, the two frames coincide on what may be learned (i.e., the messages in ϕ), but differ on what should remain unknown (i.e., whether a guess coincides with the password or has a completely different value).

The following example demonstrates resistance to guessing attacks:

Example 4.2.2. The frame $\phi = \nu pw.\{\mathsf{h}(pw)/x_1\}$ is not resistant to guessing attacks against pw because an actor who knows ϕ can guess what the password pw might be, as shown above. Formally,

$$\nu pw.\{\mathsf{h}(pw)/x_1, pw/guess\} \not\approx \nu pw, w.\{\mathsf{h}(pw)/x_1, w/guess\}.$$

Namely, denote the left frame by ϕ_1 and the right frame by ϕ_2. Then, taking $M = x_1$ and $N = \mathsf{h}(guess)$, $(M = N)\phi_1$ is true, but $(M = N)\phi_2$ is not.

However, the frame $\phi = \nu pw, n.\{\mathsf{h}(\mathsf{pair}(pw, n))\}$ is resistant to guessing attacks against pw: indeed,

$$\nu pw, n.\{\mathsf{h}(\mathsf{pair}(pw, n))/x_1, pw/guess\} \approx$$
$$\nu pw, n, w.\{\mathsf{h}(\mathsf{pair}(pw, n))/x_1, w/guess\}.$$

Intuitively, because the value of n is unknown to the actor, he cannot construct the hash to verify if a guess for pw is correct.[24] □

[24] The more general definition from the literature (cf. Corin et al. (2005), Delaune et al. (2008)) would show that the actor can *simultaneously* guess pw and n; we do not consider this generalisation here.

Using the same intuition as above, we can also define resistance to guessing attacks in the rule-based model of Chapter 3. In that

model, we would express knowledge of a hashed password with $\mathcal{C} = \{\mathsf{h}(pw|_u^\pi)\}$. As above, the password cannot be derived: $\mathcal{C} \vdash\!\!\!\cdot^{-} pw|_u^\pi$ does not hold. This is because the contents of the password do not occur elsewhere in the knowledge base. However, we can model guessing the password by adding the correct guess $guess|_\cdot^\cdot$ to the knowledge base, i.e., a new context item with contents equal to the password. Then, from knowledge base $\mathcal{C} \cup \{guess|_\cdot^\cdot\}$, $pw|_u^\pi$ and $guess|_\cdot^\cdot$ are equatable, i.e., given this knowledge base, the actor knows that the guess corresponds to the actual password. Based on this observation, we define resistance to guessing attacks in the rule-based model as follows:

Definition 4.2.3. Let $\mathfrak{c} \in \mathbb{P}^{\mathrm{cnt}}$ be a contents item, and let $\mathcal{C}$ be a knowledge base. Let $\mathcal{C}^{\mathfrak{c}}$ denote the *augmented knowledge base* $\mathcal{C} \cup \{guess|_\cdot^\cdot\}$ where $\tau(\sigma(guess|_\cdot^\cdot)) = \mathfrak{c}$[25]. We say that $\mathcal{C}$ is *resistant to guessing attacks against* $\mathfrak{c}$ if there is no p such that $\mathcal{C} \vdash\!\!\!\cdot\, \mathsf{p} \doteq guess|_\cdot^\cdot$. [26]

[25] Technically, this is knowledge with respect to an "augmented" Information Model in which context item $guess|_\cdot^\cdot$ is added. For simplicity, we leave this implicit.

[26] An equivalent definition is that, for no $\mathsf{p} \doteq guess|_\cdot^\cdot$, p is detectable. Also, we could define resistance against guessing attacks for context items rather than contents items. The above definition was chosen because it more closely resembles the equational definition, which facilitates our comparison below.

The following example demonstrates resistance to guessing attacks in the rule-based model.

Example 4.2.4. Consider the two examples in the equational setting from Example 4.2.2. We show that the same conclusions apply in the rule-based setting.

The knowledge base $\mathcal{C} = \{\mathsf{h}(pw|_u^\pi)\}$ is not resistant to guessing attacks against $\mathfrak{pw} = \tau(\sigma(pw|_u^\pi))$. Namely, as indicated above, $\mathcal{C}^{\mathfrak{pw}} \vdash\!\!\!\cdot\, pw|_u^\pi \doteq guess|_\cdot^\cdot$: the actor can apply reconstruction using the contents of $guess|_\cdot^\cdot$ to $\mathsf{h}(pw|_u^\pi)$, thus finding $pw|_u^\pi$.

On the other hand, if $pw|_u^\pi \neq n|_\cdot^\pi$, then $\mathcal{C} = \{\mathsf{h}(\{pw|_u^\pi, n|_\cdot^\pi\})\}$ is resistant to guessing attacks against $\mathfrak{pw}$. Indeed, in $\mathcal{C}^{\mathfrak{pw}}$, the actor does not know the contents of $n|_\cdot^\pi$, so he cannot derive $pw|_u^\pi$ or $n|_\cdot^\pi$ from the hash. Hence, $\mathcal{C}^{\mathfrak{pw}} \not\vdash\!\!\!\cdot\, pw|_u^\pi \doteq guess|_\cdot^\cdot$ and $\mathcal{C}^{\mathfrak{pw}} \not\vdash\!\!\!\cdot\, n|_\cdot^\pi \doteq guess|_\cdot^\cdot$. □

Above, we have defined resistance to guessing attacks given a frame of message contents in the equational setting, and given a set of context messages in the rule-based setting. We have argued that intuitively, these two definitions answer the same question: namely, given the contents of a piece of information (the guess), if it is known whether these contents occur in the messages known by the actor. In the equational setting, resistance to guessing attacks is defined in terms of static equivalence of frames with a right and wrong guess. In the rule-based setting, it is defined in terms of equatability in a knowledge base with the right guess.

Below, we use the above intuitive correspondence to define detectability and linkability in the equational setting. In the rule-based setting, detectability and linkability follow from equatability. By the above remarks, equatability in the rule-based setting intuitively corresponds to static equivalences in the equational setting. Then, in the equational setting, we simply *define* equatability in terms of these static equivalences. The result is a definition of equatability in the equational setting (and consequently, of detectability and linkability) that intuitively corresponds to the rule-based definition.

4.3 View from an Equational Knowledge Base

In this section, we propose an alternative way of determining an actor's view that exploits the expressive power of equational theories. In Chapter 3, we computed the view of an actor based on a knowledge base in the rule-based setting. As in that chapter, we model pieces of information by an Information Model I (Definition 3.1.1). However, instead of using a rule-based signature with construction/elimination rules, we now model messages using an equational signature and operations using equations. The view of an actor is then defined in terms of deducibility and static equivalence of their contents, using the intuition from the previous section.

First, we formalise messages in the equational setting. Consider an Information Model $I = (\mathsf{P}^{\mathrm{ctx}}, \mathcal{P}^{\mathrm{inf}}, \mathbb{P}^{\mathrm{cnt}}, \Leftrightarrow, \sigma, \tau)$, and let Σ^{eq} be a signature with associated equational theory E. Denote by $\mathcal{T}_{\mathsf{P}^{\mathrm{ctx}}}$ the set of *context ground terms* built from context items in $\mathsf{P}^{\mathrm{ctx}}$ using function symbols in Σ^{eq}. Such terms represent messages communicated in protocols. As in Chapter 3, an *equational knowledge base* $\mathcal{C}^{\mathrm{eq}}$ is a finite set of messages $\mathcal{C}^{\mathrm{eq}} \subset \mathcal{T}_{\mathsf{P}^{\mathrm{ctx}}}$ representing the knowledge of an actor.

We reason about contents of context ground terms using the equational theory. Namely, let $\mathcal{T}_{\mathbb{P}^{\mathrm{cnt}}}$ be the set of *contents ground terms* built from contents of pieces of information from $\mathbb{P}^{\mathrm{cnt}}$. Given a context ground term m, let $\tau(\sigma(\mathsf{m}))$ denote the term in $\mathcal{T}_{\mathbb{P}^{\mathrm{cnt}}}$ obtained from m by replacing each context item $\mathsf{p} \in \mathsf{P}^c$ by its contents $\tau(\sigma(\mathsf{p}))$. Context ground terms $\mathsf{m}_1, \mathsf{m}_2$ are called *content equivalent*, denoted $\mathsf{m}_1 \doteq \mathsf{m}_2$, if $\tau(\sigma(\mathsf{m}_1)) =_{\mathsf{E}} \tau(\sigma(\mathsf{m}_2))$.

The following example demonstrates context ground terms, equational knowledge bases, and content equivalence:

Example 4.3.1. Consider the equational signature Σ^{eq} and equational theory E from Examples 4.1.1 and 4.1.2:

$$\Sigma^{\mathrm{eq}} = \{\mathsf{pair}/2, \mathsf{enc}/2, \mathsf{pk}/1, \mathsf{penc}/3, \mathsf{fst}/1, \mathsf{snd}/1, \mathsf{dec}/2, \mathsf{pdec}/2\};$$
$$\mathsf{E} = \{\mathsf{fst}(\mathsf{pair}(x,y)) = x, \mathsf{snd}(\mathsf{pair}(x,y)) = y, \mathsf{dec}(\mathsf{enc}(x,y),y) = x,$$
$$\mathsf{pdec}(\mathsf{penc}(x,\mathsf{pk}(y),z),y) = x\}.$$

Then $shakey|^{\pi}_{\cdot}$ and $\mathsf{enc}(k|^{\pi}_{\cdot}, secret|^{\pi}_{u})$ are context ground terms, and $\mathcal{C}^{\mathrm{eq}} = \{\mathsf{enc}(secret|^{\pi}_{u}, k|^{\pi}_{\cdot}), shakey|^{\pi}_{\cdot}\}$ is an equational knowledge base. Also, $\mathsf{dec}(\mathsf{enc}(secret|^{\pi}_{u}, k|^{\pi}_{\cdot}), shakey|^{\pi}_{\cdot})$ is a context ground term; if $k|^{\pi}_{\cdot} \doteq shakey|^{\pi}_{\cdot}$, then it is content equivalent to $secret|^{\pi}_{u}$. □

To define an actor's view based on his known messages, we want to use the equivalence-based properties of deducibility and static equivalence. These properties are defined in terms of frames capturing the contents of these known messages. Hence, to use these properties, we need to consider the contents of messages. For this, define *the frame corresponding to an equational knowledge base* $\mathcal{C}^{eq}$, denoted $\phi(\mathcal{C}^{\mathrm{eq}})$, to contain the contents of all context ground terms in $\mathcal{C}^{\mathrm{eq}}$. Because we assume no a priori knowledge of actors about certain pieces of information, we define all contents items occurring

in the frame to be restricted. That is, let $\mathcal{C}^{\mathrm{eq}} = \{\mathsf{m}_1, \ldots, \mathsf{m}_k\}$, and let $n_1, \ldots, n_l \in \mathbb{P}^{\mathrm{cnt}}$ be all content items occurring in messages in $\mathcal{C}^{\mathrm{eq}}$, then:[27]

$$\phi(\mathcal{C}^{\mathrm{eq}}) := \nu n_1, \ldots, n_l.\{\tau(\sigma(\mathsf{m}_1))/x_1, \ldots, \tau(\sigma(\mathsf{m}_k))/x_k\}.$$

[27] Variables x_i can be chosen arbitrarily as long as they are distinct.

The following example shows the frame corresponding to an equational knowledge base:

Example 4.3.2. Consider the equational knowledge base $\mathcal{C}^{\mathrm{eq}} = \{\mathsf{enc}(\mathit{secret}|_u^\pi, k|_\cdot^\pi), \mathit{shakey}|_\cdot\}$ from Example 4.3.1; suppose[28]

$$\tau(\sigma(k|_\cdot^\pi)) = \tau(\sigma(\mathit{shakey}|_\cdot)) = c_1 \neq c_2 = \tau(\sigma(\mathit{secret}|_u^\pi)).$$

[28] For our framework, the exact contents of pieces of information are irrelevant: the only thing that matters is the content equivalence relation $\doteq$ on them. Therefore, we can identify pieces of information with symbols $c_1, c_2, \ldots$

Then

$$\phi(\mathcal{C}^{\mathrm{eq}}) = \nu c_1, c_2.\{\mathsf{enc}(c_2, c_1)/x_1, c_1/x_2\}$$

is the frame corresponding to this equational knowledge base. □

We now show how to define equatability of context items using static equivalence of frames. As in the rule-based setting, we will use equatability to define both the detectability and the associability component of an actor's view. To define equatability, we use the intuition of the preceding section. In particular, consider any equational knowledge base $\mathcal{C}^{\mathrm{eq}}$ and its corresponding frame $\phi(\mathcal{C}^{\mathrm{eq}})$. Let p be a context item, $[\mathsf{p}]$ its equivalence set under content equivalence, and $\mathsf{P} \subset [\mathsf{p}]$. Moreover, let $\phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$ be the frame obtained from $\phi(\mathcal{C}^{\mathrm{eq}})$ by replacing the contents corresponding to items in P by some fresh new name w. For instance, in the above example, $\phi_{\{k|_\cdot^\pi\}}(\mathcal{C}^{\mathrm{eq}}) = \nu c_1, c_2, w.\{\mathsf{enc}(c_2, w)/x_1, c_1/x_2\}$. Note that, if $\mathsf{P} = \varnothing$ or $\mathsf{P} = [\mathsf{p}]$, then $\phi(\mathcal{C}^{\mathrm{eq}}) \approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$. Conversely, if $\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$, then there is some equation $M = N$ that holds in $\phi(\mathcal{C}^{\mathrm{eq}})$ but that fails to hold when we replace some items (namely, those in P) in $[\mathsf{p}]$ by w, but not others. We interpret M and N as a "check" telling the actor that some context item $\mathsf{p} \in \mathsf{P}$ replaced by w is content equivalent to some non-replaced context item $\mathsf{p}' \in [\mathsf{p}] \setminus \mathsf{P}$. Note that this is a generalisation of our remarks on resistance to guessing attacks from the previous section: there, we had $\mathsf{P} = \{\mathit{guess}|_\cdot\}$ and noted that non-static equivalence of $\phi(\mathcal{C}^{\mathrm{eq}+\mathfrak{c}})$ and $\phi_{\{\mathit{guess}|_\cdot\}}(\mathcal{C}^{\mathrm{eq}+\mathfrak{c}})$ should correspond to equatability of $\mathit{guess}|_\cdot$ to some other context item in its equivalence set under $\doteq$.

We illustrate our intuition with the following example:

Example 4.3.3. Consider equational knowledge base

$$\mathcal{C}^{\mathrm{eq}} = \{\mathsf{enc}(d_1|_u^\pi, k_1|_\cdot^\pi), \mathsf{enc}(d_2|_u^\pi, k_2|_\cdot^\pi), d_3|_u^\pi, k_1|_\cdot^\pi\}$$

(where $d_1|_u^\pi \doteq d_2|_u^\pi \doteq d_3|_u^\pi$) with corresponding frame

$$\phi(\mathcal{C}^{\mathrm{eq}}) = \nu d, k_1, k_2.\{\mathsf{enc}(d, k_1)/x_1, \mathsf{enc}(d, k_2)/x_2, d/x_3, k_1/x_4\}.$$

Then

$$\phi_{\{d_1|_u^\pi\}}(\mathcal{C}^{\mathrm{eq}}) = \nu d, k_1, k_2, w.\{\mathsf{enc}(w, k_1)/x_1, \mathsf{enc}(d, k_2)/x_2, d/x_3, k_1/x_4\},$$

and $\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\{d_1|_u^\pi\}}(\mathcal{C}^{\mathrm{eq}})$: indeed, $\mathsf{enc}(x_3, x_4) = x_1$ holds in $\phi(\mathcal{C}^{\mathrm{eq}})$ but not in $\phi_{\{d_1|_u^\pi\}}(\mathcal{C}^{\mathrm{eq}})$. Intuitively, this equation tells us that $d_1|_u^\pi \doteq d_3|_u^\pi$ (with $d_1|_u^\pi \in \{d_1|_u^\pi\}$ and $d_3|_u^\pi \notin \{d_1|_u^\pi\}$). Namely, in $\mathcal{C}^{\mathrm{eq}}$, we can encrypt $d_3|_u^\pi$ with $k_1|^\pi$ and obtain a message with the same contents as $\mathsf{enc}(d_1|_u^\pi, k_1|^\pi)$, which means that the plaintext $d_1|_u^\pi$ of $\mathsf{enc}(d_1|_u^\pi, k_1|^\pi)$ must have the same contents as $d_3|_u^\pi$.

On the other hand,

$$\phi_{\{d_2|_u^\pi\}}(\mathcal{C}^{\mathrm{eq}}) = \nu d, k_1, k_2, w.\{\mathsf{enc}(d, k_1)/x_1, \mathsf{enc}(w, k_2)/x_2, d/x_3, k_1/x_4\},$$

and $\phi(\mathcal{C}^{\mathrm{eq}}) \approx \phi_{\{d_2|_u^\pi\}}$. Indeed, because the actor does not have decryption key k_2, he has no way to see if $d_2|_u^\pi \in \{d_2|_u^\pi\}$ is content equivalent to any context item in $\{d_1|_u^\pi, d_3|_u^\pi\}$. □

To respect the above intuition, we want to define equatability $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq \mathsf{p}'$ in terms of static equivalence $\approx$ so that:

$$\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}}) \text{ iff } \exists \mathsf{p} \in \mathsf{P}, \mathsf{p}' \notin \mathsf{P} : \mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq \mathsf{p}'. \quad (\dagger)$$

We now provide an explicit definition for equatability; we show next that this definition respects (†), and moreover, that it is the only equivalence relation that does so.

Definition 4.3.4. Let $\mathcal{C}^{\mathrm{eq}}$ be an equational knowledge base. Let $[\mathsf{p}]$ denote the equivalence class of context item p under content equivalence. The *equatability* relation on context items, denoted $\mathcal{C}^{\mathrm{eq}} \vdash * \doteq *$, is defined by the following rule:

$$\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq \mathsf{p}' \text{ iff } \forall \mathsf{P} \subset [\mathsf{p}] \text{ s.t. } \mathsf{p} \in \mathsf{P}, \mathsf{p}' \notin \mathsf{P} : \phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}}).$$

The next two lemmas show that Definition 4.3.4 indeed defines an equivalence relation; and that it is the only equivalence relation that respects (†).

Lemma 4.3.5. The relation $\mathcal{C}^{\mathrm{eq}} \vdash * \doteq *$, as defined in Definition 4.3.4, is an equivalence relation.

Proof. Reflexivity is clear (for $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq \mathsf{p}$, the condition for the $\forall$ quantifier cannot be satisfied).

For symmetry, note that

$$\phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}}) \approx \phi_{[\mathsf{p}] \setminus \mathsf{P}}(\mathcal{C}^{\mathrm{eq}}). \quad (4.1)$$

This is true because the two frames are equal up to swapping the two restricted names w and $\tau(\sigma(\mathsf{p}))$, hence they satisfy the same equations. Now, suppose $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq \mathsf{p}'$. We need to show $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p}' \doteq \mathsf{p}$. Hence, take any $\mathsf{P} \subset [\mathsf{p}']$ such that $\mathsf{p}' \in \mathsf{P}$, $\mathsf{p} \notin \mathsf{P}$. Because $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq \mathsf{p}'$,

$$\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{[\mathsf{p}] \setminus \mathsf{P}}(\mathcal{C}^{\mathrm{eq}}).$$

Combining this with (4.1), we get $\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$, as we needed to show.

Finally, for transitivity, suppose $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p}_1 \doteq \mathsf{p}_2$ and $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p}_2 \doteq \mathsf{p}_3$. We need to show $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p}_1 \doteq \mathsf{p}_3$. Hence, take any $\mathsf{P} \subset [\mathsf{p}_1]$ such that

$\mathsf{p}_1 \in \mathsf{P}$, $\mathsf{p}_3 \notin \mathsf{P}$. We need to show that $\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$. But if $\mathsf{p}_2 \notin \mathsf{P}$, then we have a P such that $\mathsf{p}_1 \in \mathsf{P}$, $\mathsf{p}_2 \notin \mathsf{P}$, hence $\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$ follows from $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p}_1 \doteq \mathsf{p}_2$. On the other hand, if $\mathsf{p}_2 \in \mathsf{P}$, then we have a P such that $\mathsf{p}_2 \in \mathsf{P}$, $\mathsf{p}_3 \notin \mathsf{P}$, hence $\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$ follows from $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p}_2 \doteq \mathsf{p}_3$.

This shows that we have an equivalence relation. □

Lemma 4.3.6. Let $\mathcal{C}^{\mathrm{eq}} \vdash * \doteq' *$ denote any equivalence relation on context items such that, for all sets P of content equivalent context items:

$$\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}}) \text{ iff } \exists \mathsf{p} \in \mathsf{P}, \mathsf{p}' \notin \mathsf{P} : \mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq' \mathsf{p}'.$$

Then the relation $\mathcal{C}^{\mathrm{eq}} \vdash * \doteq' *$ coincides with $\mathcal{C}^{\mathrm{eq}} \vdash * \doteq *$.

Proof. ($\subset$): First, assume that $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq' \mathsf{p}'$; we show that $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq \mathsf{p}'$. Namely, take any $\mathsf{P} \subset [\mathsf{p}]$ such that $\mathsf{p} \in \mathsf{P}, \mathsf{p}' \notin \mathsf{P}$. We need to show that $\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$. Clearly, $\exists \mathsf{p} \in \mathsf{P}, \mathsf{p}' \notin \mathsf{P} : \mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq' \mathsf{p}'$ holds, so $\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$ follows by assumption.

($\supset$): Now, assume that $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq \mathsf{p}'$, i.e.,

$$\forall \mathsf{P} \subset [\mathsf{p}] \text{ s.t. } \mathsf{p} \in \mathsf{P}, \mathsf{p}' \notin \mathsf{P} : \phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}}). \qquad (\ddagger)$$

We need to show $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq' \mathsf{p}'$. Let $\mathsf{p}_0 = \mathsf{p}$, $\mathsf{P}_0 = \{\mathsf{p}_0\}$. By ($\ddagger$), we have $\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}_0}(\mathcal{C}^{\mathrm{eq}})$. By definition, $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p}_0 \doteq' \mathsf{p}_1$ for some p_1 in $[\mathsf{p}] \setminus \mathsf{P}_0$. Either $\mathsf{p}_1 = \mathsf{p}'$, in which case we are done, or we proceed: let $\mathsf{P}_1 = \{\mathsf{p}_0, \mathsf{p}_1\}$. By ($\ddagger$), we have $\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\mathsf{P}_1}(\mathcal{C}^{\mathrm{eq}})$; by assumption, there exists $\mathsf{p}'' \in \mathsf{P}_1$, $\mathsf{p}_2 \notin \mathsf{P}_1$ such that $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p}'' \doteq' \mathsf{p}_2$. In fact, because $\mathcal{C}^{\mathrm{eq}} \vdash * \doteq' *$ is an equivalence relation and all items in P_1 are mutually equatable: $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p}_i \doteq' \mathsf{p}_2$ for all $i \in \{0, 1\}$. Again, if $\mathsf{p}_2 = \mathsf{p}'$, we are done; otherwise, consider $\mathsf{P}_2 = \{\mathsf{p}_0, \mathsf{p}_1, \mathsf{p}_2\}$. This construction either extends the collection of sets P_i, or gives $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq' \mathsf{p}'$. However, for all P_i we have $\mathsf{P}_i \subset [\mathsf{p}]$ and $[\mathsf{p}]$ is finite, so at one point it is no longer possible to extend the collection. Hence, the construction eventually gives $\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq' \mathsf{p}'$. □

We now show how to use Definition 4.3.4 to determine equatability.

Example 4.3.7. Consider the equational knowledge base

$$\mathcal{C}^{\mathrm{eq}} = \{i|_u^\kappa, \mathsf{enc}(i|_u^\kappa, k|_\cdot^\kappa), \mathsf{enc}(i|_u^\pi, k|_\cdot^\pi), \mathsf{enc}(i|_u^\alpha, l|_\cdot^\alpha)\}$$

where

$$\tau(\sigma(i|_u^\pi)) = \tau(\sigma(i|_u^\kappa)) = \tau(\sigma(i|_u^\alpha)) = i;$$
$$\tau(\sigma(k|_\cdot^\kappa)) = \tau(\sigma(k|_\cdot^\pi)) = k; \quad \tau(\sigma(l|_\cdot^\alpha)) = l.$$

We determine which context items in $\{i|_u^\pi, i|_u^\kappa, i|_u^\alpha\}$ are equatable to each other. By Definition 4.3.4:

$$\mathcal{C}^{\mathrm{eq}} \vdash i|_u^\pi \doteq i|_u^\kappa \text{ iff } \phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\{i|_u^\pi\}}(\mathcal{C}^{\mathrm{eq}}) \wedge \phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\{i|_u^\pi, i|_u^\alpha\}}(\mathcal{C}^{\mathrm{eq}}).$$

Now:

$$\phi(\mathcal{C}^{\mathrm{eq}}) = \nu i,k,l.\{i/x_1, \mathsf{enc}(i,k)/x_2, \mathsf{enc}(i,k)/x_3, \mathsf{enc}(i,l)/x_4\};$$
$$\phi_{\{i|_u^\pi\}}(\mathcal{C}^{\mathrm{eq}}) = \nu i,k,l,w.\{i/x_1, \mathsf{enc}(i,k)/x_2, \mathsf{enc}(w,k)/x_3, \mathsf{enc}(i,l)/x_4\};$$
$$\phi_{\{i|_u^\pi, i|_u^\alpha\}}(\mathcal{C}^{\mathrm{eq}}) = \nu i,k,l,w.\{i/x_1, \mathsf{enc}(i,k)/x_2, \mathsf{enc}(w,k)/x_3, \mathsf{enc}(w,l)/x_4\}.$$

One can verify that

$$\phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\{i|_u^\pi\}}(\mathcal{C}^{\mathrm{eq}}); \phi(\mathcal{C}^{\mathrm{eq}}) \not\approx \phi_{\{i|_u^\pi, i|_u^\alpha\}}(\mathcal{C}^{\mathrm{eq}}),$$

so indeed, $i|_u^\pi$ and $i|_u^\kappa$ are equatable. However, $i|_u^\alpha$ is not equatable to $i|_u^\pi$ or $i|_u^\kappa$: indeed, $\phi(\mathcal{C}^{\mathrm{eq}}) \approx \phi_{\{i|_u^\alpha\}}(\mathcal{C}^{\mathrm{eq}})$, where

$$\phi_{\{i|_u^\alpha\}}(\mathcal{C}^{\mathrm{eq}}) = \nu i,k,l,w.\{i/x_1, \mathsf{enc}(i,k)/x_2, \mathsf{enc}(i,k)/x_3, \mathsf{enc}(w,l)/x_4\}.$$

Note that the rule-based model from Chapter 3 would have given the same results. □

Using equatability, we can finally define the view following from an equational knowledge base. In our rule-based definition (Definition 3.5.3), associability was defined completely in terms of equatability, so we can easily re-define it in the equational setting. For detectability, however, there is one final twist. Namely, detectability is defined in terms not just of equatability, but also of derivability. However, for derivability, we have not defined an equivalent in the equational setting yet. We fix this by noting that, given a context item $\mathsf{p} \in \mathsf{P}^{\mathrm{ctx}}$, we do have a way to see if its contents are known: this is simply checking deducibility $\phi(\mathcal{C}^{\mathrm{eq}}) \vdash \tau(\sigma(\mathsf{p}))$. Now, consider the augmented equational knowledge base $\mathcal{C}^{\mathrm{eq}+\tau(\sigma(\mathsf{p}))}$ obtained by adding a context item $guess|_\cdot$ with contents $\tau(\sigma(\mathsf{p}))$ to $\mathcal{C}^{\mathrm{eq}}$. Now, p should be detectable if and only if the contents of p' are known and $\mathsf{p} \doteq guess|_\cdot$. (It is straightforward to see that this is also true in the rule-based model.) This gives a definition of detectability in terms of deducibility and static equivalence. We obtain the following definition of view:

Definition 4.3.8. Let $\mathcal{C}^{\mathrm{eq}} \subset \mathcal{T}_{\mathsf{P}^c}$ be an equational knowledge base. The *view corresponding to* $\mathcal{C}^{eq}$ is the view $V = (\mathsf{O}, \leftrightarrow)$ such that:

- The set $\mathsf{O} \subset \mathsf{P}^{\mathrm{ctx}}$ of detectable items consists of those $\mathsf{p} \in \mathsf{P}^{\mathrm{ctx}}$ for which $\phi(\mathcal{C}^{\mathrm{eq}}) \vdash \tau(\sigma(\mathsf{p}))$ and $\mathcal{C}^{\mathrm{eq}+\tau(\sigma(\mathsf{p}))} \vdash \mathsf{p} \doteq guess|_\cdot$;
- The associability relation $\leftrightarrow$ is the least equivalence relation on contexts in O^c such that, whenever $\mathcal{C}^{\mathrm{eq}} \vdash i|_u^\pi \doteq i'|_l^\eta$ for context identifiers $i|_u^\pi, i'|_l^\eta$, then $*|_u^\pi \leftrightarrow *|_l^\eta$.

The following example shows the view corresponding to an equational knowledge base.

Example 4.3.9. Consider the equational knowledge base

$$\mathcal{C}^{\mathrm{eq}} = \{i|_u^\kappa, \mathsf{enc}(i|_u^\kappa, \mathsf{enc}(i|_u^\pi, k|_\cdot^\pi), k|_\cdot^\kappa), \mathsf{enc}(i|_u^\alpha, l|_\cdot^\alpha)\}$$

from Example 4.3.7. We determine the view $V = (\mathsf{O}, \leftrightarrow)$ corresponding to $\mathcal{C}^{\mathrm{eq}}$.

We first find out which of the identifiers $i|_u^\pi$, $i|_u^\kappa$, $i|_u^\alpha$ are detectable, i.e., elements of O. For this, we need to find out whether their contents are deducible and whether they are equatable to $guess|_\cdot^\cdot$ in the augmented equational knowledge base $\mathcal{C}^{\mathrm{eq}+i} = \mathcal{C}^{\mathrm{eq}} \cup \{guess|_\cdot^\cdot\}$ with $guess|_\cdot^\cdot \doteq i|_u^\kappa \doteq i|_u^\pi \doteq i|_u^\alpha$. Clearly, their contents are deducible, i.e., $\phi(\mathcal{C}^{\mathrm{eq}}) \vdash i$. For equatability, we proceed similarly to Example 4.3.7 to find that $i|_u^\pi$ and $i|_u^\kappa$ are equatable to $guess|_\cdot^\cdot$, hence detectable; but $i|_u^\alpha$ is not.

The associability relation $\leftrightarrow$ follows from the equatability of the identifiers as determined in Example 4.3.7; hence, $*|_u^\pi \leftrightarrow *|_u^\kappa$ is the only known association.

Again, note that this view coincides with the view we would have obtained using the rule-based approach. □

The following example shows that using equational knowledge bases, we can verify detectability both with and without the assumption of visible failure; hence overcoming one of the limitations of the rule-based model.

Example 4.3.10. Consider equational knowledge base

$$\mathcal{C}^{\mathrm{eq}} = \{k|_\cdot^\cdot, \mathsf{enc}(n|_\cdot^\pi, k|_{srv}^\pi)\}$$

with $\tau(\sigma(k|_\cdot^\cdot)) = \tau(\sigma(k|_{srv}^\pi)) = k$; $\tau(\sigma(n|_\cdot^\pi)) = n$. When using equational theory E from Example 4.1.2, $k|_{srv}^\pi$ is detectable: intuitively, the actor learn $k|_{srv}^\pi$ by observing that decryption of $\mathsf{enc}(n|_\cdot^\pi, k|_{srv}^\pi)$ using $k|_\cdot^\cdot$ succeeds. However, when using equational theory E' introduced in Example 4.1.8 to model encryption without visible failure, $k|_{srv}^\pi$ is no longer detectable because he cannot recognise the result $n|_\cdot^\pi$. Formally, consider frames

$$\phi = \phi(\mathcal{C}^{\mathrm{eq}+k}) = \nu k, i, j.\{k/x_1, \mathsf{enc}(n,k)/x_2, k/x_3\};$$
$$\phi' = \phi_{\{k|_{srv}^\pi\}}(\mathcal{C}^{\mathrm{eq}+k}) = \nu k, i, j, w.\{k/x_1, \mathsf{enc}(n,w)/x_2, k/x_3\}.$$

Using equational theory E, we have that $\mathsf{enc}(\mathsf{dec}(x_2,x_1),x_1)\phi =_{\mathsf{E}} x_2\phi$ but $\mathsf{enc}(\mathsf{dec}(x_2,x_1),x_1)\phi' \neq_{\mathsf{E}} x_2\phi'$: re-encryption with the correct key gives the same result but re-encryption with a wrong key does not. Hence $\phi \not\approx \phi'$, giving detectability of $k|_{srv}^\pi$. However, using equational theory E', both $\mathsf{enc}(\mathsf{dec}(x_2,x_1),x_1)\phi =_{\mathsf{E}'} x_2\phi$ and $\mathsf{enc}(\mathsf{dec}(x_2,x_1),x_1)\phi' =_{\mathsf{E}'} x_2\phi'$ so the frames cannot be distinguished in this way. In fact, $\phi \approx \phi'$, so $k|_{srv}^\pi$ is not detectable.

On the other hand, if we consider equational knowledge base

$$\{k|_\cdot^\cdot, \mathsf{enc}(n|_\cdot^\pi, k|_{srv}^\pi), n|_\cdot^\pi\},$$

then $k|_{srv}^\pi$ is detectable both using equational theory E and using equational theory E'. Namely, in both equational theories, the actor can recognise the result of decrypting $\mathsf{enc}(n|_\cdot^\pi, k|_{srv}^\pi)$ by comparing it to known message $n|_\cdot^\pi$. Formally,

$$\nu k, i, j.\{k/x_1, \mathsf{enc}(n,k)/x_2, n/x_3, k/x_4\} \not\approx$$
$$\nu k, i, j, w.\{k/x_1, \mathsf{enc}(n,w)/x_2, n/x_3, k/x_4\} :$$

for instance, $\mathsf{dec}(x_2, x_1) = x_3$ holds in the left frame (in this frame, x_2 is an encryption of x_3 under key x_1) but not in the right frame (in this frame, it is not). □

4.4 *Rule-Based vs Equational Model*

In the previous section, we showed how to compute an actor's view from an equational knowledge base (Definition 4.3.8), intuitively generalising our previous definition (Definition 3.5.3). Indeed, using this definition, we can determine views also when the assumptions on cryptographic primitives in the rule-based setting do not hold. However, it is of course desirable that if the assumptions *do* hold, then the rule-based and equational definitions are the same. In this section, we show that indeed, using a suitable translation from construction/elimination rules to equations, the definitions coincide. That is, the equational model for detectability and associability is really a generalisation of the rule-based model.

We now specify the translation from the rule-based model to the equational model, and state our main result. Let Σ be a signature in the rule-based model with associated set of construction and elimination rules. We assume that each function symbol $\mathsf{f}/k \in \Sigma$ in the rule-based signature only has the standard construction rule $\mathsf{f}(x_1, \ldots, x_k) \leftarrow x_1, \ldots, x_k$; as discussed in Section 3.8, this is not a stringent restriction. We now construct an equational signature Σ^{eq} and equational theory E^{eq}. Namely, Σ^{eq} consists of all function symbols $\mathsf{f}/k \in \Sigma$; and of one function symbol $\mathsf{d}/(k+1) \in \Sigma^{\mathrm{eq}}$ for every elimination rule with k auxiliary messages. In this case, we call d the *name* of the elimination rule. The equational theory E^{eq} consists of an equation for each elimination rule; namely, given rule

$$\mathrm{fn}(m_1, \ldots, m_k) \xrightarrow{\doteq n_1, \ldots, \doteq n_l} r$$

with name d, add equation

$$\mathsf{d}(\mathrm{fn}(m_1, \ldots, m_k), n_1, \ldots, n_l) = r. \qquad (\circ)$$

In particular, we add function symbols and equations for every elimination rule introduced due to the visible failure assumption. Adding function symbols and equations for reconstruction rules is not needed.

Because each function symbol in Σ is also in Σ^{eq}, messages in the rule-based setting using signature Σ are also context ground terms using equational signature Σ^{eq}. In particular, a knowledge base using signature Σ in the rule-based setting can also be seen as an equational knowledge base using equational signature Σ^{eq}. The main result we prove in this section is that detectability and linkability in Σ coincide with detectability and linkability in Σ^{eq}:

Theorem 4.4.1. Let $\mathcal{C}$ be a knowledge base using signature Σ. Let Σ^{eq} be the equational signature corresponding to Σ, as defined above. Then the views corresponding to $\mathcal{C}$ in the rule-based model

(Definition 3.5.3) and in the equational model (Definition 4.3.8) coincide.

We prove the theorem in Section 4.5. The following example demonstrates the above construction.

Example 4.4.2. Consider the model for deterministic symmetric encryption, deterministic asymmetric encryption, cryptographic hash functions, and digital signatures from Section 3.4. The corresponding equational signature and equational theory for these primitives are:

$$\Sigma^{\mathrm{eq}} = \{\mathsf{enc}/2, \mathsf{dec}/2, \mathsf{tdec}/2, \mathsf{pk}/1, \mathsf{aenc}/2, \mathsf{adec}/2, \mathsf{tadec}/2, \mathsf{h}/1, \mathsf{sig}/2, \mathsf{vsig}/2, \mathsf{vsig2}/2\}$$

$$\begin{aligned} \mathsf{dec}(\mathsf{enc}(m,k),k) &= m \quad & \mathsf{tdec}(\mathsf{enc}(m,k),k) &= k \\ \mathsf{adec}(\mathsf{aenc}(m,\mathsf{pk}(k)),k) &= m \quad & \mathsf{tadec}(\mathsf{aenc}(m,\mathsf{pk}(k)),k) &= k \\ \mathsf{vsig}(\mathsf{sig}(m,k),\mathsf{pk}(k)) &= m \quad & \mathsf{vsig2}(\mathsf{sig}(m,k),\mathsf{pk}(k)) &= \mathsf{pk}(k) \end{aligned}$$

Similarly, a k-length list can be modelled using function symbols $\mathsf{lst}k$, $\mathsf{get}k1/1, \ldots, \mathsf{get}kk/1$ and equations $\mathsf{get}ki(\mathsf{lst}(x_1, \ldots, x_k)) = x_i$.

Hence, by the above theorem, for any knowledge base $\mathcal{C}$, the view corresponding to $\mathcal{C}$ using the rule-based signature from Section 3.4 and the view corresponding to $\mathcal{C}$ using the above equational signature coincide. □

The next example shows that the equational theory E^{eq} obtained using the above construction can often be simplified.

Example 4.4.3. Consider the equational theory from the above example. This equational theory can be simplified in several ways.

For frames that do not contain function symbol tdec, the equation $\mathsf{tdec}(\mathsf{enc}(m,k),k) = k$ is irrelevant both for deducibility of names and for static equivalence. For deducibility, any term $\mathsf{tdec}(x,y)$ in a recipe is either $=_{E^{\mathsf{eq}}}$-equivalent to y (so y can be used in the recipe instead), or it can be replaced, along with all $=_{E^{\mathsf{eq}}}$-equivalent terms $\mathsf{tdec}(x',y')$, by some arbitrary name without impacting the deduction. For static equivalence, for any terms x, y, z where $z \neq \mathsf{tdec}(...)$ and frame ϕ, we have that $\mathsf{tdec}(x,y)\phi =_{E^{eq}} z\phi$ if and only if $y\phi =_{E^{eq}} z\phi$ and $\mathsf{enc}(\mathsf{dec}(x,y),y)\phi =_{E^{eq}} x\phi$, hence any check that can be performed using tdec can also be performed without it. Similarly, tadec is redundant because $\mathsf{tadec}(x,y)\phi =_{E^{eq}} z\phi$ (with $z \neq \mathsf{tadec}(...)$) iff $y\phi =_{E^{eq}} z\phi$ and $\mathsf{aenc}(\mathsf{adec}(x,y),\mathsf{pk}(y))\phi =_{E^{eq}} x\phi$.

Finally, the above equational signature and equational theory are infinite because they contain k-length lists for all k. There are two possible ways to obtain a finite equational signature (which is required by existing tools). The first possibility is to remove function symbols $\mathsf{lst}k$, $\mathsf{get}ki$ if there are no k-length lists in the knowledge bases considered. The second possibility is to consider only pairs (i.e., length-2 lists), and express longer-length lists as nested pairs, e.g., $\mathsf{lst2}(\mathsf{lst2}(x,y),z)$ instead of $\mathsf{lst3}(x,y,z)$. (Note that

$$\Sigma^{\mathrm{eq}} = \{\mathsf{enc}/2, \mathsf{dec}/2, \mathsf{pk}/1, \mathsf{aenc}/2, \mathsf{adec}/2, \mathsf{h}/1, \mathsf{sig}/2, \mathsf{vsig}/2, \mathsf{vsig2}/2, \mathsf{pair}/2, \mathsf{fst}/1, \mathsf{snd}/1\}$$
$$\mathsf{E}^{eq} = \{\mathsf{dec}(\mathsf{enc}(m,k),k) = m, \mathsf{adec}(\mathsf{aenc}(m,\mathsf{pk}(k)),k) = m,$$
$$\mathsf{vsig}(\mathsf{sig}(m,k),\mathsf{pk}(k)) = m, \mathsf{vsig2}(\mathsf{sig}(m,k),\mathsf{pk}(k)) = \mathsf{pk}(k),$$
$$\mathsf{fst}(\mathsf{pair}(x,y)) = x, \mathsf{snd}(\mathsf{pair}(x,y)) = y\}$$

Figure 4.1: Signature and equational theory for standard cryptographic primitives equivalent to the rule-based model from Section 3.4 (see Example 4.4.3)

$\mathsf{lst2}(\mathsf{lst2}(x,y),z) \neq_{\mathsf{E}^{eq}} \mathsf{lst2}(x,\mathsf{lst2}(y,z))$, so the conversion of lists to pairs needs to be done consistently.)

We conclude that the equational theory shown in Figure 4.1 is equivalent to the model of deterministic symmetric encryption, deterministic asymmetric encryption, cryptographic hash functions, and digital signatures from Section 3.4. □

By telling us to which equational theory our rule-based approach corresponds, Theorem 4.4.1 provides confidence in the results of our rule-based analysis. At the same time, in some cases it allows us to re-use existing equational models in the rule-based setting. We discuss this in Section 4.7.

4.5 *Proof of Correspondence Result*

We now prove Theorem 4.4.1. The proof is in two steps. The main technical difference between the rule-based and equational models is the way they handle failed cryptographic operations. Namely, in the rule-based model, a cryptographic operation is modelled by a rule that states the result of a successful application of the operation; a failed application means that the rule does not apply. In the equational model, a cryptographic operations is modelled by a function symbol, called a *destructor*, and an equation describing its result. This means that, unlike in the rule-based model, messages using these function symbols can also represent the result of failed cryptographic operations. Thus, the first step we take towards proving Theorem 4.4.1 is to show that the modelling of failed cryptographic operations is in a sense unnecessary for deducibility and static equivalence in Σ^{eq}. The second step is then to show the correspondence result without taking into account failed cryptographic operations.

For the first step, we note that the equational theory E^{eq} can be seen as a convergent rewriting system[29]. Namely, consider the rewriting system $\mathcal{R}$ obtained by orienting the equations ($\circ$) from left to right, i.e., $\mathcal{R}$ is the set of rules $\mathsf{d}(\mathsf{fn}(m_1,\ldots,m_k),n_1,\ldots,n_l) \to r$. Let $\to_{\mathcal{R}}$ denote the rewrite relation corresponding to $\mathcal{R}$; that is, given ground terms t_1, t_2, we have $t_1 \to_{\mathcal{R}} t_2$ if there is a rule $l \to r \in \mathcal{R}$ and a substitution σ of ground terms for variables such that t_2 is obtained from t_1 by replacing an occurrence of $l\sigma$ by $r\sigma$. Let $\to^*_{\mathcal{R}}$ denote its reflexive, transitive closure. Clearly, $\mathcal{R}$ is *terminating*, i.e., there are no infinite $\to_{\mathcal{R}}$ chains (indeed, every rewrite step reduces the number of destructors in the term). Also, $\mathcal{R}$ is *locally confluent*, i.e., if $l \to_{\mathcal{R}} r_1$

[29] Cf. Ciobâcă et al. (2009)

and $l \to_{\mathcal{R}} r_2$, then there exists r such that $r_1 \to_{\mathcal{R}}^* r$ and $r_2 \to_{\mathcal{R}}^* r$. Informally, because destructors occur only at the head of rewrite rules in $\mathcal{R}$, any rule $l \to_{\mathcal{R}} r_2$ applicable to l can also be applied to r_1. If $t \to_{\mathcal{R}}^* t'$ and there is no t'' such that $t' \to_{\mathcal{R}} t''$, then we call t' a *normal form* of t, and write $t{\Downarrow_{\mathcal{R}}} = t'$. By a classical result[30], termination and local confluence imply that $\mathcal{R}$ is convergent, meaning that this normal form t' is unique. In particular, $t_1 =_{\mathsf{E}^{eq}} t_2$ iff $t_1{\Downarrow_{\mathcal{R}}} = t_2{\Downarrow_{\mathcal{R}}}$. Below, we use the equational theory E^{eq} and rewriting system $\mathcal{R}$ interchangeably.

[30] Cf. Dershowitz and Jouannaud (1990)

We now formalise the intuition that "modelling failed cryptographic operations is unnecessary". A *constructor term* is a ground term which does not contain any destructors, i.e., function symbols d coming from elimination rules ($\circ$). We call ground t a *regular deduction* if, for all subterms u of t (including t itself), $u{\Downarrow}$ is a constructor term. In particular, because of convergence, this implies that there is a series of rewriting steps

$$t = t_1 \to_{\mathcal{R}} t_2 \to_{\mathcal{R}} \ldots \to_{\mathcal{R}} t_k = t{\Downarrow},$$

where each term that is replaced is a destructor applied to constructor terms.

Intuitively, if no destructors occur in a frame, then there is no reason why applying a failed cryptographic operation would help to derive any message from the frame. The following lemma formalises that intuition:

Lemma 4.5.1. Let Σ be a rule-based signature, and Σ^{eq} the corresponding equational signature. Let ϕ be a frame consisting of only constructor terms, and let t be a constructor term. Then $\phi \vdash t$ if and only there is a recipe M such that $M\phi$ is a regular deduction, and $M\phi{\Downarrow} = t$.

Proof. ($\Leftarrow$) is trivial. For ($\Rightarrow$), we show that, for any recipe M such that $M\phi{\Downarrow} = t$, there exists a recipe M' with $M\phi{\Downarrow} = t$ so that $M\phi$ is a regular deduction. We do this by induction on the number of non-applicable destructors in M, i.e., the number of subterms d of M such that $d = \mathsf{d}(M, N_1, \ldots, N_l)$ and $d\phi{\Downarrow} = \mathsf{d}(M\phi{\Downarrow}, N_1\phi{\Downarrow}, \ldots, N_l\phi{\Downarrow})$.

The base case $k = 0$ is clear. Namely, in this case, M has no such subterms and ϕ only contains constructor terms, so $M\phi$ is itself a regular deduction and we are done.

Now, suppose that the requested property holds for all M'' with $\leq k$ such subterms, and that M has $k+1$ such subterms. Take any such subterm N, and let M^- be the recipe in which we replace all maximal subterms N' of M such that $N'\phi{\Downarrow} = N\phi{\Downarrow}$ by some arbitrary name x. All rewrite steps $M\phi \to_{\mathcal{R}} M\phi{\Downarrow}$ that do not occur inside one of the replaced subterms still apply, giving us a sequence $M^-\phi \to_{\mathcal{R}}^* M^-\phi{\Downarrow}$ of rewriting steps. Moreover, because M^- is obtained from M by consistently replacing subterms and no subterms of t were replaced (because t is a constructor term), $M^-\phi{\Downarrow} = M\phi{\Downarrow}$. Finally, M^- contains $\leq k$ non-applicable destructors, so by induction there is an M' such that $M'\phi$ is a regular deduction and $M'\phi{\Downarrow} = M^-\phi{\Downarrow} = M\phi{\Downarrow}$, which is what we wanted to show. □

We now characterise static equivalence in terms of regular deductions. Recall that static equivalence means that the same equations between recipes hold in the two frames. Analogously to above, we now just compare equations between recipes *that give rise to regular deductions*. However, this is not enough: due to visible failure, the fact that a recipe gives a regular deduction in one frame but not in the other allows an actor to distinguish the frames. For instance, consider

$$\phi_1 = \nu m.\{\mathsf{enc}(m,k)/x_1\}, \phi_2 = \nu m.\{\mathsf{enc}(m,l)/x_1\}.$$

These frames are not statically equivalent because $\mathsf{tdec}(x_1,k) = k$ holds in ϕ_1 but not in ϕ_2; however, $\mathsf{tdec}(x_1,k) = k$ is not a regular deduction in ϕ_2 so the above criterion does not hold. Note that in this case, no recipes that give regular deductions in both frames can distinguish them. However, if we additionally demand that the same regular deductions exist in both frames, then we can prove the following correspondence:

Lemma 4.5.2. Let Σ be a rule-based signature, and Σ^{eq} the corresponding equational signature. Let ϕ, ψ be two frames that consist only of constructor terms and that do not contain unrestricted names[31]. Suppose that ϕ and ψ are equal up to names, i.e., for each variable x, $x\phi$ and $x\psi$ have the same term structure using the same function symbols, but possibly different names[32]. Then $\phi \approx \psi$ if and only if:

1. For all recipes M for ϕ and ψ, $M\phi$ is a regular deduction iff $M\psi$ is a regular deduction;
2. For all recipes M, N for ϕ and ψ such that $M\phi$, $N\phi$ are regular deductions, $M\phi =_{\mathsf{E}^{eq}} N\phi$ iff $M\psi =_{\mathsf{E}^{eq}} N\psi$.

Proof. ($\Rightarrow$). First assume $\phi \approx \psi$.

For the first property, suppose that $M\phi$ is a regular deduction. We show, by induction on the complexity of recipe M, that $M\psi$ is also a regular deduction; and that $M\phi\Downarrow$ and $M\psi\Downarrow$ are equal up to names.

The base case, i.e., M is a name or variable, is clear.

First, let $M = \mathsf{f}(M_1, \ldots, M_k)$, where f is not a destructor, and suppose that the requested property holds for all subterms of f. In this case, it clearly also holds for f.

Now, suppose that $M = \mathsf{d}(N, N_1, \ldots, N_l)$ where d is a destructor, and that the requested property holds for all subterms of M. Suppose that $M\phi$ is a regular deduction. We need to show that $M\psi$ is also a regular deduction, and that $M\phi\Downarrow$ and $M\psi\Downarrow$ are equal up to names.

Because $M\phi$ is a regular deduction, $N\phi, N_1\phi, \ldots, N_l\phi$ are regular deductions; and hence by induction, $N\psi, N_1\psi, \ldots, N_l\psi$ are regular deductions and give terms that are equal up to names. Suppose that d comes from elimination rule

$$\mathsf{fn}(m_1, \ldots, m_k) \xrightarrow{\doteq n_1, \ldots, \doteq n_l} r.$$

[31] The result could be generalised also to frames with unrestricted names by considering renamings of restricted variables in case they clash with unrestricted ones. For simplicity and because we do not need the more general result, we do not consider this here.

[32] We need that frames are equal up to names for technical reasons. Intuitively, this lemma holds because Σ^{eq} models cryptographic operations that satisfy visible failure: if failure can be observed anyway, then there is no need to work with the result of a failed operation. However, Σ^{eq} does not model visible failure for operations *with no auxiliary messages*. By demanding that frames only differ up to names, we ensure that operations with no auxiliary messages succeed in ϕ if and only if they succeed in ψ so this does not cause difficulty. The result holds more generally if "test equations" for such operations are added to Σ^{eq}.

Because $M\phi$ is a regular deduction, the corresponding rewrite rule $\mathsf{d}(\mathsf{fn}(m_1,\ldots,m_k),n_1,\ldots,n_l) \to_{\mathcal{R}} r$ applies to $\mathsf{d}(N\phi\Downarrow,N_1\phi\Downarrow,\ldots,N_l\phi\Downarrow)$.

First consider the case $l = 0$, i.e., the elimination rule has no auxiliary messages. Note that variables in $\mathsf{fn}(m_1,\ldots,m_k)$ are used at most once and that, by induction, $N\phi\Downarrow, N_i\phi\Downarrow$ and $N\psi\Downarrow, N_i\psi\Downarrow$ are equal up to names. This means that the rewrite rule corresponding to this elimination rule also applies to $\mathsf{d}(N\psi\Downarrow,N_1\psi\Downarrow,\ldots,N_l\psi\Downarrow)$, and that it gives the same term up to names. Hence $M\psi$ is a regular deduction that give the same result as $M\phi$ up to names.

Now, consider the case $l > 0$. Because of visible failure, rewrite rule $\mathsf{d}'(\mathsf{fn}(m_1,\ldots,m_k),n_1,\ldots,n_l) \to_{\mathcal{R}} n_1$ with some name d' exists. Because $\mathsf{d}(\mathsf{fn}(m_1,\ldots,m_k),n_1,\ldots,n_l) \to_{\mathcal{R}} r$ applies to $\mathsf{d}(N\phi\Downarrow,N_1\phi\Downarrow,\ldots,N_l\phi\Downarrow)$, $\mathsf{d}'(\mathsf{fn}(m_1,\ldots,m_k),n_1,\ldots,n_l) \to_{\mathcal{R}} n_1$ applies to $\mathsf{d}'(N\phi\Downarrow,N_1\phi\Downarrow,\ldots,N_l\phi\Downarrow)$. Hence, $\mathsf{d}'(N,N_1,\ldots,N_l)\phi\Downarrow = N_1\phi\Downarrow$.

Because ϕ and ψ are statically equivalent, also $\mathsf{d}'(N,N_1,\ldots,N_l)\Downarrow\psi = N_1\psi\Downarrow$; hence rewrite rule $\mathsf{d}'(\mathsf{fn}(m_1,\ldots,m_k),n_1,\ldots,n_l) \to_{\mathcal{R}} n_1$ applies to $\mathsf{d}'(N\psi\Downarrow,N_1\psi\Downarrow,\ldots,N_l\psi\Downarrow)$; hence also d applies, meaning that $M\psi$ is also a regular deduction giving the same result as $M\phi$ up to names. This completes our proof by induction.

The second property is true by definition of static equivalence.

($\Leftarrow$). Assume that the two given properties hold.

We need to show that, for all recipes M, N with $M\phi =_{\mathsf{E}^{eq}} N\phi$, we have $M\psi =_{\mathsf{E}^{eq}} N\psi$. We proceed by induction on the number m of subterms t of M, N such that $t\phi$ is not a regular deduction. If $m = 0$, i.e., there are no such subterms, then we are done by the second property.

Now, suppose that the requested property holds for all pairs of recipes with $\leq m$ subterms that do not correspond to regular deductions, and suppose that M, N have $m+1$ such subterms. Take a minimal subterm u of M, N such that $u\phi$ is not a regular deduction. Then $u = \mathsf{d}(M', M'_1,\ldots,M'_l)$ with $u\phi\Downarrow = \mathsf{d}(M'\phi\Downarrow, M'_1\phi\Downarrow,\ldots,M'_l\phi\Downarrow)$. Note that by minimality, $M'\phi, M'_1\phi,\ldots,M'_l\phi$ are all regular deductions. Hence, by the first property, $M'\psi, M'_1\psi,\ldots,M'_l\psi$ are regular deductions. Moreover, because $u\phi\Downarrow$ is not a regular deduction, neither is $u\psi\Downarrow$. We conclude that $u\psi\Downarrow = \mathsf{d}(M'\psi\Downarrow, M'_1\psi\Downarrow,\ldots,M'_l\psi\Downarrow)$.

Now, consider the set U of all minimal subterms u' of M, N such that $u'\phi =_{\mathsf{E}^{eq}} u\phi$. Take any $u' = \mathsf{f}(N', N'_1,\ldots,N'_k) \in U$; again, $N'\phi$, $N'_1\phi$, $\ldots$, $N'_l\phi$ are all regular deductions. By the same line of reasoning as above, $u'\psi\Downarrow = \mathsf{d}(N'\psi\Downarrow, N'_1\psi\Downarrow,\ldots,N'_l\psi\Downarrow)$. Moreover, by the first property, because $M'\phi\Downarrow = N'\phi\Downarrow$, also $M'\psi\Downarrow = N'\psi\Downarrow$, and similarly, $M'_i\psi\Downarrow = N'_i\psi\Downarrow$. We conclude that, for all $u' \in U$, $u'\psi =_{\mathsf{E}^{eq}} u\psi$.

Finally, let M^u, N^u be obtained from M, N by replacing all $u' \in U$ by some fresh name x. Following the rewrite steps for $M\phi =_{\mathsf{E}^{eq}} N\phi$ that do not happen inside these subterms, we conclude $M^u\phi =_{\mathsf{E}^{eq}} N^u\phi$. Moreover, M^u, N^u contain $\leq m$ subterms that are not regular deductions. Hence, by induction, $M^u\psi =_{\mathsf{E}^{eq}} N^u\psi$. Now, because $u\psi =_{\mathsf{E}^{eq}} u'\psi$ for all $u, u' \in U$, we can substitute back the respective us in M^u, N^u to get $M\psi =_{\mathsf{E}^{eq}} N\psi$, as requested. □

Having characterised deducibility and static equivalence in terms of regular deductions, we now prove Theorem 4.4.1 by establishing a link between derivations in our deductive system and regular deductions. Namely, we describe derivations in our deductive systems as recipes operating at the context layer: we show how derivations correspond to context-layer recipes, and how context-layer recipes correspond to the content-layer recipes used in deduciblity and static equivalence.

Derivations in our deductive system correspond to *message recipes*: recipes that use only variables x_i corresponding to messages, and no names. Informally, the application of a construction rule in our deductive system corresponds to the use of a non-destructor symbol in a recipe; the application of an elimination rule corresponds to the use of a destructor. More formally, identify each message m_i in a knowledge base $\mathcal{C}$ with the variable x_i it gets in its frame $\phi(\mathcal{C})$, then derivations and message recipes relate as follows:

- Application of ($\vdash\!\!\!\cdots\!\!\!\overline{\ }\,\mathbf{0}$) to derive m_i corresponds to recipe x_i;
- If derivations for $\mathsf{n}_1, \ldots, \mathsf{n}_l$ correspond to recipes $x_1, \ldots, x_l$, then applying rule ($\vdash\!\!\!\cdots\mathbf{C}$) corresponds to recipe $\mathsf{f}(x_1, \ldots, x_l)$;
- If derivations for $\mathsf{m}, \mathsf{n}_1, \ldots, \mathsf{n}_k$ correspond to recipes $x, x_1, \ldots, x_k$, then applying rule ($\vdash\!\!\!\cdots\mathbf{E}$) using elimination rule d corresponds to recipe $\mathsf{d}(x, x_1, \ldots, x_k)$.

For instance, the derivation from Figure 3.3 (page 46) corresponds to message recipe

$$\mathsf{vsig2}(\mathsf{get22}(\mathsf{dec}(x_8, x_5)), x_6)$$

(using message numbers according to the order given in Example 3.5.4, and names for elimination rules as in Example 4.4.2).

Given this correspondence between derivations and message recipes, the result of a derivation can be described by a rewriting system analogously to the one on context ground terms. Namely, recipe M acts on a knowledge base in the obvious way: $M\mathcal{C}$ is obtained by substituting context messages in $\mathcal{C}$ for variables in M. Then, define rewrite relation $\to$ on context terms as follows: $\mathsf{m} \to \mathsf{n}$ if there is an elimination rule

$$\mathsf{f}(x_1, \ldots, x_k) \xrightarrow{\doteq y_1, \ldots, \doteq y_l} z,$$

with name d, substitution σ of context messages[33] for variables, and m is obtained by replacing $\mathsf{d}(\mathsf{f}(\mathsf{m}_1, \ldots, \mathsf{m}_k), \mathsf{n}_1, \ldots, n_l)$ by n, where $\mathsf{m}_i = x_i\sigma$; $\mathsf{n} = z\sigma$; and $\mathsf{n}_i \doteq y_i\sigma$. Then $\to^*$ is the reflexive, transitive closure of $\to$. If $M\mathcal{C} \to^* \mathsf{m}$ and m is a context message, we write $\mathsf{m} = M\mathcal{C}\Downarrow$. For instance, proceeding with the above example, we get the sequence of rewrite steps shown in Figure 4.2.

[33] This ensures that an elimination rule is applied after the elimination rules for its submessages

It is straightforward to see that the following lemma holds:

Lemma 4.5.3. Let $\mathcal{C}$ be a knowledge base using signature Σ. Let Σ^{eq} be the equational signature corresponding to Σ, as defined in Section 4.4. Then:

$$\begin{aligned}\mathsf{vsig2}(\mathsf{get22}(\mathsf{dec}(x_8,x_5)),x_6)\mathcal{C} &= \mathsf{vsig2}(\mathsf{get22}(\mathsf{dec}(\mathsf{enc}(\{id|_{su},\mathsf{sig}(\{age|_{su},n|.\},k^-|_{srv})\},shkey|.)|^{\pi},skey|^{\cdot}_{\cdot})),\mathsf{pk}(k^-|^{\cdot}_{srv}))\\ &\to \mathsf{vsig2}(\mathsf{get22}(\{id|_{su},\mathsf{sig}(\{age|_{su},n|.\},k^-|_{srv})\}|^{\pi}),\mathsf{pk}(k^-|^{\cdot}_{srv}))\\ &\to \mathsf{vsig2}(\mathsf{sig}(\{age|_{su},n|.\},k^-|_{srv})|^{\pi}),\mathsf{pk}(k^-|^{\cdot}_{srv}))\\ &\to \mathsf{pk}(k^-|^{\pi}_{srv})\end{aligned}$$

Figure 4.2: Rewrite steps corresponding to derivation of Figure 3.3

- $\mathcal{C} \vdash \mathsf{m}$ if and only if there exists a message recipe M such that $M\mathcal{C} \to^* \mathsf{m}$;
- If $M\mathcal{C} \to^* \mathsf{m}$, then $M\phi(\mathcal{C})$ is a regular deduction with $M\phi(\mathcal{C})\Downarrow = \tau(\sigma(\mathsf{m}))$;
- If $M\phi(\mathcal{C})$ is a regular deduction, then $M\mathcal{C} \to^* \mathsf{m}$ for some m with $\tau(\sigma(\mathsf{m})) = M\phi(\mathcal{C})\Downarrow$.

We now compare equatability in the rule-based and equational settings. Recall that equatability in the equational setting is defined using static equivalences. As a first step towards linking equatability in the two settings, we show that equatability in the rule-based setting corresponds to these static equivalences. Intuitively, we show that equatability in the rule-based setting satisfies the intuitive relation (†) (page 69).

In fact, instead of proving (†), we prove a slightly more general statement. Recall that equatability in the equational model is defined in terms of static equivalence of frame $\phi(\mathcal{C}^{\mathrm{eq}})$ to frames $\phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$ obtained by mapping particular context items to a fresh new name instead of their contents. We can interpret this operation in terms of our three-layer model by saying that $\phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$ is the frame corresponding to $\mathcal{C}^{\mathrm{eq}}$ in an alternative Information Model in which the contents of items in P have been replaced by fresh contents. We now prove a general result that links equatability in a knowledge base $\mathcal{C}$ to static equivalences of frames with respect to different Information Models. Below, let us we write $\mathcal{C} \vdash^I \mathsf{p}$, $\mathcal{C} \vdash^I \mathsf{p} \doteq \mathsf{p}'$, $\phi^I(\mathcal{C})$, etcetera to make explicit with respect to which Information Model I a certain operation is defined. The following result holds:

Lemma 4.5.4. Let $\mathcal{C}$ be a knowledge base using signature Σ. Let Σ^{eq} be the equational signature corresponding to Σ, as defined in Section 4.4. Let I, I' be two Information Models containing all context items in $\mathcal{C}$. Then $\phi^I(\mathcal{C}) \not\approx \phi^{I'}(\mathcal{C})$ if and only if there exist $\mathsf{p}, \mathsf{p}' \in \mathsf{P}^{\mathrm{ctx}}$ such that:

$$((\mathcal{C} \vdash^I \mathsf{p} \doteq_0 \mathsf{p}') \wedge \mathsf{p} \not\doteq^{I'} \mathsf{p}') \vee ((\mathcal{C} \vdash^{I'} \mathsf{p} \doteq_0 \mathsf{p}') \wedge \mathsf{p} \not\doteq^{I} \mathsf{p}').$$

(Here, $\mathcal{C} \vdash *$ is according to the rule-based definition using Σ, and $\approx$ is according to the equational theory for equational signature Σ^{eq}.)

Proof. ($\Leftarrow$) Let p, p' be context items such that $\mathcal{C} \vdash^I \mathsf{p} \doteq_0 \mathsf{p}'$ but $\mathsf{p} \not\doteq^{I'} \mathsf{p}'$. We show that $\phi^I(\mathcal{C}) \not\approx \phi^{I'}(\mathcal{C})$. (The other case is analogous.) We do this by contradiction: so, suppose that in fact, $\phi^I(\mathcal{C}) \approx \phi^{I'}(\mathcal{C})$.

Because $\mathcal{C} \vdash^I \mathsf{p} \doteq_0 \mathsf{p}'$, there are m, m' such that $\mathcal{C} \vdash^I \mathsf{m}$, $\mathcal{C} \vdash^I \mathsf{m}'$, $\mathsf{m}@z = \mathsf{p}$, $\mathsf{m}'@z = \mathsf{p}'$. Let M, M' be the message recipes such that $M\mathcal{C} \to_I^* \mathsf{m}$, $M'\mathcal{C} \to_I^* \mathsf{m}'$. Then, by Lemma 4.5.3, $M\phi(\mathcal{C}), M'\phi(\mathcal{C})$ are regular deductions such that $M\phi^I(\mathcal{C}) =_{\mathsf{E}^{eq}} M'\phi^I(\mathcal{C})$. By static equivalence, $M\phi^{I'}(\mathcal{C}), M'\phi^{I'}(\mathcal{C})$ are regular deductions such that $M\phi^{I'}(\mathcal{C}) =_{\mathsf{E}^{eq}} M'\phi^{I'}(\mathcal{C})$. Note that the result of rewrite relation $\to$, if it exists, is independent from the information and contents layers. Hence, $M\mathcal{C} \to_{I'}^* \mathsf{m}$ and $M' \to_{I'}^* \mathsf{m}'$. However, because $M\phi^I(\mathcal{C}) =_{\mathsf{E}^{eq}} M'\phi^I(\mathcal{C})$ we must have $\mathsf{m} \doteq_{I'} \mathsf{m}'$, whereas we know that $\mathsf{m}@z = \mathsf{p} \not\doteq^{I'} \mathsf{m}'@z$. Contradiction, so the two frames cannot be statically equivalent.

($\Rightarrow$) Suppose that $\phi^I(\mathcal{C}) \not\approx \phi^{I'}(\mathcal{C})$. We need to find p, p' such that $\mathcal{C} \vdash^I \mathsf{p} \doteq_0 \mathsf{p}'$ and $\mathsf{p} \not\doteq^{I'} \mathsf{p}'$, or the same with I and I' exchanged. By Lemma 4.5.2, non-static-equivalence means that one of the two conditions given in that lemma must fail to hold.

Suppose that the second condition does not hold, i.e., there exist M, N such that $M\phi^I(\mathcal{C})$, $M\phi^{I'}(\mathcal{C})$, $N\phi^I(\mathcal{C})$ and $N\phi^{I'}(\mathcal{C})$ are all regular deductions; and that $M\phi^I(\mathcal{C}) =_{\mathsf{E}^{eq}} N\phi^I(\mathcal{C})$ but $M\phi^{I'}(\mathcal{C}) \neq_{\mathsf{E}^{eq}} N\phi^{I'}(\mathcal{C})$ (the other possibility, namely equality in I' and inequality in I, is handled analogously). Because $M\phi^I(\mathcal{C})$, $M\phi^{I'}(\mathcal{C})$ are regular deductions, we have $M\mathcal{C} \to_I^* \mathsf{m}$, $M\mathcal{C} \to_{I'}^* \mathsf{m}$ (as above, they must give the same message because the result of rewrite rules $\to$ is determined only by the context layer). Equally, $N\mathcal{C} \to_I^* \mathsf{n}$, $N\mathcal{C} \to_{I'}^* \mathsf{n}$. Also, $\mathsf{m} \doteq^I \mathsf{n}$, but $\mathsf{m} \not\doteq^{I'} \mathsf{n}$. Hence, there is z such that $\mathsf{m}@z, \mathsf{n}@z \in \mathsf{P}^c$, $\mathsf{m}@z \doteq^I \mathsf{n}@z$ and $\mathsf{m}@z \not\doteq^{I'} \mathsf{n}@z$. Hence, $\mathcal{C} \vdash^I \mathsf{m}@z \doteq_0 \mathsf{n}@z$, as requested.

Suppose now that the first condition does not hold. Suppose there exists M such that $M\phi^I(\mathcal{C})$ is a regular deduction, but $M\phi^{I'}(\mathcal{C})$ is not. (Again, the other possibility is handled analogously.) Take a minimal such M, i.e., $M = \mathsf{d}(N, N_1, \ldots, N_l)$ such that $N\phi^I(\mathcal{C})$, $N_i\phi^I(\mathcal{C})$, $N\phi^{I'}(\mathcal{C})$, $N_i\phi^{I'}(\mathcal{C})$ are all regular deductions; and rewrite rule $\mathsf{d}(\mathsf{f}(x_1, \ldots, x_k), y_1, \ldots, y_l) \to z$ applies to $\mathsf{d}(N\phi^I(\mathcal{C})\Downarrow, N_1\phi^I(\mathcal{C})\Downarrow, \ldots, N_l\phi(\mathcal{C}^I)\Downarrow)$, but not to $\mathsf{d}(N\phi^{I'}(\mathcal{C})\Downarrow, N_1\phi^{I'}(\mathcal{C})\Downarrow, \ldots, N_l\phi^{I'}(\mathcal{C})\Downarrow)$. By Lemma 4.5.3, the rewrite rule applies to $\mathsf{d}(N\mathcal{C}\Downarrow, N_1\mathcal{C}\Downarrow, \ldots, N_l\mathcal{C}\Downarrow)$ using Information Model I but not using Information Model I'.

Recall that the substitution of context messages for variables in an elimination rule, and hence in the above rewrite rule, is completely determined by the message $\mathsf{f}(x_1, \ldots, x_k)$. Let σ be the substitution such that $\mathsf{f}(x_1, \ldots, x_k)\sigma = N\mathcal{C}\Downarrow$. Apparently, for some q, $N_q\mathcal{C}\Downarrow \doteq^I y_q\sigma$, but $N_q\mathcal{C}\Downarrow \not\doteq^{I'} y_q\sigma$. Now, consider the rewrite rule $\mathsf{d}'(\mathsf{f}(x_1, \ldots, x_k), y_1, \ldots, y_l) \to y_q$ that exists due to the visible failure assumption. Then this rule applies to $\mathsf{d}'(N\mathcal{C}\Downarrow, N_1\mathcal{C}\Downarrow, \ldots, N_l\mathcal{C}\Downarrow)$, giving result $y_q\sigma$. On the other hand, recipe N_q gives $N_q\mathcal{C}\Downarrow$. Because $N_q\mathcal{C}\Downarrow \doteq^I y_q\sigma$ but $N_q\mathcal{C}\Downarrow \not\doteq^{I'} y_q\sigma$, for some z we must have that $N_q\mathcal{C}\Downarrow@z = \mathsf{p}$ and $y_q\sigma@z = \mathsf{p}'$, where $\mathsf{p} \not\doteq^{I'} \mathsf{p}'$. Hence, $\mathcal{C} \vdash^I \mathsf{p} \doteq_0 \mathsf{p}'$ but $\mathsf{p} \not\doteq^{I'} \mathsf{p}'$, as requested. □

We now apply the above lemma to obtain a link between equatability in the rule-based and equational settings:

Lemma 4.5.5. Let $\mathcal{C}$ be a knowledge base using signature Σ. Let Σ^{eq} be the equational signature corresponding to Σ, as defined in Section 4.4. Then:

$$\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p} \doteq \mathsf{p}' \text{ iff } \mathcal{C} \vdash \mathsf{p} \doteq \mathsf{p}'.$$

(Here, $\vdash\!\!\!\cdot$ is with respect to Σ, and $\vdash$ is with respect to Σ^{eq}.)

Proof. ($\Rightarrow$) Suppose that $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p} \doteq \mathsf{p}'$. By definition, we need to show that for all $\mathsf{P} \subset [\mathsf{p}]$ such that $\mathsf{p} \in \mathsf{P}, \mathsf{p}' \notin \mathsf{P}'$, we have $\phi(\mathcal{C}) \not\approx \phi_{\mathsf{P}}(\mathcal{C})$. Let I be the Information Model for $\mathcal{C}$, so $\phi(\mathcal{C}) = \phi^{I}(\mathcal{C})$. Let I' be the Information Model obtained by replacing the contents of all context items in P by some fresh new value, so $\phi_{\mathsf{P}}(\mathcal{C}) = \phi^{I'}(\mathcal{C})$. Because $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p} \doteq \mathsf{p}'$, there exist $\mathsf{p}_1 \in \mathsf{P}, \mathsf{p}_2 \notin \mathsf{P}$ such that $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p}_1 \doteq_0 \mathsf{p}_2$. Furthermore, by definition of I', $\mathsf{p}_1 \not\doteq^{I'} \mathsf{p}_2$. Hence, by Lemma 4.5.4 we have

$$\phi(\mathcal{C}) = \phi^{I}(\mathcal{C}) \not\approx \phi^{I'}(\mathcal{C}) = \phi_{\mathsf{P}}(\mathcal{C}),$$

which is what we needed to show.

($\Leftarrow$) Suppose that $\mathcal{C} \vdash \mathsf{p} \doteq \mathsf{p}'$. Taking $\mathsf{P} = \{\mathsf{p}\}$, by definition we have $\phi(\mathcal{C}) \not\approx \phi_{\mathsf{P}}(\mathcal{C})$. As above, let I be the Information Model for $\mathcal{C}$, and let I' be the Information Model such that $\phi_{\mathsf{P}}(\mathcal{C}) = \phi^{I'}(\mathcal{C})$. By Lemma 4.5.4, we have p'' such that $\mathcal{C} \vdash\!\!\!\cdot^{I}\; \mathsf{p} \doteq_0 \mathsf{p}''$ and $\mathsf{p} \not\doteq^{I'} \mathsf{p}''$. (By construction of I and I', the converse is not possible.) If $\mathsf{p}' = \mathsf{p}''$, we are done. Otherwise, continuing with $\mathsf{P} = \{\mathsf{p}, \mathsf{p}''\}$, we find p''' such that $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p} \doteq_0 \mathsf{p}'''$ or $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p}'' \doteq_0 \mathsf{p}'''$, so $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p} \doteq \mathsf{p}'''$. Continuing, we eventually find $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p} \doteq \mathsf{p}'$ (cf. proof of Lemma 4.3.6). □

Now, Theorem 4.4.1 follows:

Proof (of Theorem 4.4.1). Let $V^{r} = (\mathsf{O}^{r}, \leftrightarrow^{r})$ and $V^{eq} = (\mathsf{O}^{eq}, \leftrightarrow^{eq})$ be the views corresponding to $\mathcal{C}$ according to Definitions 3.5.3 and 4.3.8, respectively.

By Lemma 4.5.5, equatability in the rule-based and equational settings coincide. Because the associability relation is defined entirely in terms of equatability, clearly, $\leftrightarrow^{r}$ and $\leftrightarrow^{eq}$ coincide.

Now, suppose that $\mathsf{p} \in \mathsf{O}^{r}$. By definition, there is $\mathsf{p}' \in \mathsf{P}^{\mathrm{ctx}}$ such that $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p}'$ and $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p} \doteq \mathsf{p}'$. By Lemma 4.5.3, we have $\phi(\mathcal{C}) \vdash \tau(\sigma(\mathsf{p}'))$, with $\tau(\sigma(\mathsf{p}')) = \tau(\sigma(\mathsf{p}))$. Moreover, $\mathcal{C}^{\tau(\sigma(\mathsf{p}))} \vdash\!\!\!\cdot\; \mathsf{p} \doteq \mathsf{p}'$ and $\mathcal{C}^{\tau(\sigma(\mathsf{p}))} \vdash\!\!\!\cdot\; \mathsf{p}' \doteq_0 \mathit{guess}|^{:}_{:}$, so by transitivity $\mathcal{C}^{\tau(\sigma(\mathsf{p}))} \vdash\!\!\!\cdot\; \mathsf{p} \doteq \mathit{guess}|^{:}_{:}$, so by Lemma 4.5.5, $\mathcal{C}^{\tau(\sigma(\mathsf{p}))} \vdash \mathsf{p} \doteq \mathit{guess}|^{:}_{:}$. Hence, $\mathsf{p} \in \mathsf{O}^{eq}$.

Conversely, suppose that $\mathsf{p} \in \mathsf{O}^{eq}$. Then $\phi(\mathcal{C}) \vdash \tau(\sigma(\mathsf{p}))$ and $\mathcal{C}^{\tau(\sigma(\mathsf{p}))} \vdash \mathsf{p} \doteq \mathit{guess}|^{:}_{:}$. By Lemma 4.5.3, there exists $\mathsf{p}' \doteq \mathsf{p}$ such that $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p}'$. By Lemma 4.5.5, $\mathcal{C}^{\tau(\sigma(\mathsf{p}))} \vdash\!\!\!\cdot\; \mathsf{p} \doteq \mathit{guess}|^{:}_{:}$. Because $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p}'$, we can repeat the reasoning used to conclude $\mathcal{C}^{\tau(\sigma(\mathsf{p}))} \vdash\!\!\!\cdot\; \mathsf{p} \doteq \mathit{guess}|^{:}_{:}$ in $\mathcal{C}$ to obtain $\mathcal{C} \vdash\!\!\!\cdot\; \mathsf{p} \doteq \mathsf{p}'$. Hence, $\mathsf{p} \in \mathsf{O}^{r}$.

Hence, $V^{r} = V^{eq}$, as we wanted to show.

□

4.6 Implementation

Our tool for formal analysis of privacy in communication protocols[34] includes a proof-of-concept implementation for computing

[34] Available at `http://code.google.com/p/objective-privacy/`

views using equational theories. Intuitively, it translates each detectability query to one deducibility query and a number of static equivalence queries based on Definitions 4.3.8 and 4.3.4. Similarly, it translates each associability query to a number of static equivalence queries based on Definitions 4.3.8 and 4.3.4. These queries are then evaluated by the KiSS tool[35] for computing knowledge of actors under equational theories. Finally, our tools interprets the deducibility and static equivalence results from KiSS as detectability and associability results.

[35] Ciobâcă et al. (2009)

We have not optimised this implementation for efficiency; in particular, whereas detectability requires a frame to be non-statically-equivalent to *all* frames in a certain set[36], the implementation will continue evaluating the remainder of the static equivalences even if it finds that one of them holds. As a consequence, the computations of views that took much less than a second in the rule-based setting (see Section 3.7) at present typically take several minutes in the equational setting. We leave optimisation of our implementation in the equational setting as future work.

[36] Of size exponential in the number of items content equivalent to the item at hand

4.7 Discussion

Using Existing Equational Theories Existing formalisations of cryptographic primitives can be used with the approach of this chapter. For instance, the models of standard cryptographic primitives given in this chapter are well-known. Note that some formalisations of cryptographic primitives in the literature are not suitable for verifying static equivalence, as needed for our approach. Namely, when formalising cryptographic primitives for the verification of secrecy and correspondence properties[37], it is unnecessary to model randomness[38]. Such formalisations without randomness do not give accurate results when verifying static equivalence. On the other hand, formalisations used for verifying privacy properties based on observational equivalence[39] can all be used for our approach.

[37] See, e.g., Blanchet and Smyth (2011)

[38] Cortier et al. (2007)

[39] E.g., Dahl et al. (2011), Delaune et al. (2009), Dong et al. (2012)

Many cryptographic primitives have been modelled using equational theories; for instance, bit-commitments and blind signatures[40]; non-interactive zero-knowledge protocols[41]; and encryption schemes satisfying various security assumptions[42]. In general, these formal models are idealisations of the underlying cryptographic primitives, and hence cannot provide complete assurance that obtained results apply in practice. However, for some particular models of basic cryptographic primitives, instantiations by actual cryptographic schemes are known[43] for which static equivalence guarantees privacy properties in the more accurate "computation model" of cryptography.

[40] Delaune et al. (2009)

[41] Backes et al. (2008)

[42] Corin et al. (2005)

[43] E.g., Baudet et al. (2010), Backes et al. (2008)

Apart from equational models of cryptographic primitives, also models using "destructors"[44] can be used with our approach. Destructors, used in the popular ProVerif[45] tool for protocol verification, model cryptographic operations using rewrite rules rather than equational theories. Static equivalence can also be defined in this

[44] Blanchet et al. (2008)

[45] Blanchet and Smyth (2011)

setting[46], and verified using ProVerif. Alternatively, these rewrite rules can be interpreted as equations for verification with tools like KiSS[47]; it can be shown[48] that, because our approach only requires the verification of static equivalences of frames that differ up to names, this gives the same results.

[46] Blanchet et al. (2008)

[47] Ciobâcă et al. (2009)

[48] This is almost a direct consequence of consequence of Lemma 4.5.2

Translations Between Models By using the correspondence between equational and rule-based models (Theorem 4.4.1), we can re-use existing equational formalisations of primitives in our rule-based model. This is useful because the rule-based model, while less expressive, is much more efficient to evaluate. As a simple example, "bit commitment"[49] can be modelled with equation $open(commit(m,r),r) = m$[50]; reasoning along the lines of Example 4.4.3 shows that we can equivalently use a rule-based model with function symbol commit/2 and elimination rule

$$\mathsf{commit}(x,y) \xrightarrow{\doteq y} x.$$

[49] A bit commitment binds the actor constructing it to a choice of either 0 or 1 without immediately revealing that choice; the choice is later revealed by "opening" the commitment

[50] Delaune et al. (2009)

Similar remarks also apply to destructor-based models of cryptographic primitives.

If a precise correspondence is not possible, it may still be possible to safely approximate equations in the rule-based model. For instance, consider the equational theory for encryption without visible failure (Example 4.1.8). It can be seen easily that dropping equation $\mathsf{enc}(\mathsf{dec}(m,k),k) = m$ gives the attacker strictly more power if frames never contain the dec symbol. Hence, our rule-based mode of encryption, which corresponds to the model without that equation, can be seen as a safe approximation of this equational model.

Conversely, rule-based formalisations of cryptographic primitives can also be re-used in the equational model to verify equivalence-based privacy properties. The following cautionary remark is in place. The equational theory corresponding to a rule-based model satisfies the visible failure assumptions for all cryptographic operations that take auxiliary messages. However, for operations that do not take auxiliary messages, this it not necessarily the case. This does not make a difference for our correspondence theorem because it only concerns frames that are the same up to names, in which case the attacker cannot learn anything from the (non-)applicability of such an operation. However, it may make a difference for other privacy properties. If so desired, visible failure can be modelled by adding a "test equation" $\mathsf{test}(m) = \mathsf{ok}$ for each elimination rule $m \to n$. Rule-based formalisations can also be converted to destructors by interpreting equations ($\circ$) as rewrite rules; in this case, operations without auxiliary messages *do* satisfy visible failure.

5
Symbolic Verification of Detectability and Associability

Contents

THE METHODOLOGY DEVELOPED SO FAR has allowed us to get an accurate picture of the knowledge of the actors in the scenario at the beginning of Chapter 2. Unfortunately, it is not clear how general this picture is. For instance, what happens if the message sent from Alice to Bob contains not Steve's date of birth, but the city he lives in? What if Eve can guess the passport number inside the encryption? *Intuitively*, it is clear what will happen: Bob and the coalition of Eve and Mallory will learn Steve's city, and the knowledge of the passport number does not help Eve to decipher the message. *Formally*, however, such small changes in the scenario means that the whole analysis as presented in the previous chapter needs to be re-done.

In this chapter, we develop a generalisation of the framework presented in Chapters 2–3 that allows us to draw conclusions about knowledge of actors *symbolically*, i.e., independently from the actual information transmitted in a particular scenario. Rather than reasoning on the information exchanged in a given scenario, we reason abstractly about what conditions a scenario should satisfy for knowledge to be derived. Importantly, this allows us to reason also about how privacy can be broken by external knowledge from protocols that we have not modelled. For instance, this reasoning tells us that both Bob and the coalition of Eve and Mallory can learn the contents of the encryption in any case; and that Eve learns the contents of the forwarded message only if some other protocols leaks the decryption key that is used (in particular, the knowledge of the passport

does not help).

Although this chapter introduces a new model of messages and knowledge, this new model has a strong connection to the model of Chapter 3. We re-use the formalisation of cryptographic primitives from that chapter; and we show a very precise connection between the reasoning in this chapter and that in Chapter 3. For instance, in this chapter, we may conclude that Eve learns the contents of the forwarded message if the decryption key is leaked. We prove that this conclusion translates exactly to the setting of Chapter 3, i.e., we can conclude that for all instantiated models and knowledge bases in which the decryption key (in that instantiation) is leaked, the contents of the forwarded message (in that instantiation) is learned.

On the other hand, we also show how to perform a symbolic analysis of privacy without considering the instantiated model of Chapter 3. Namely, we propose a visualisation of symbolic conclusions about personal information, and present decision procedures showing how privacy guarantees can be read from this visualisation that are guaranteed to be true regardless of any particular scenario. This way, we are able to obtain privacy guarantees with more assurance than in our previous models.

Outline In this chapter:

- We generalise our instantiated model of (non-)personal information, messages, and protocols to a symbolic model that is independent from actual information (§5.1);
- We introduce constraints as a way of expressing conditions under which knowledge can be derived (§5.2);
- We show how to determine constraints for derivability of messages (§5.3), and equatability of pieces of information (§5.4);
- We show how to visualise the constraints relevant for knowledge of personal information in a constraint graph, and how to use this graph to analyse privacy (§5.5);
- We discuss our algorithm to automatically determine constraints for derivability (§5.6);
- We generalise the symbolic model, so that it can deal with messages containing arbitrary numbers of pieces of information (§5.7);
- We discuss possible improvements to the approach (§5.8).

5.1 Information, Messages, and Protocols

In this section, we model information in communication protocols independently from any particular protocol instance. Recall that a context item $v|_l^\pi$ models a piece of information by specifying the protocol instance π in which it occurs, the role l in the protocol instance

that the information refers to, and the particular piece information v about role l in instance π that is represented. If domain η represents another instance of the same protocol as π, then also item $v|_l^\eta$ exists, but it generally represents a different piece of information.

The idea of our symbolic model is to use "symbolic items" $v|_l$ to reason about the piece of information in *any* instance of the protocol. For instance, we will draw conclusions like "$v|_l$ is derivable if $k|_l$ is derivable", meaning that "for any protocol instance π, $v|_l^\pi$ is derivable if $k|_l^\pi$ is derivable", etcetera. We say that $v|_l^\pi$ and $v|_l^\eta$ are *instantiations* of $v|_l$. As with context items, we distinguish symbolic non-personal items, data items, and identifiers. We explicitly mark symbolic items as *random* (meaning that the information they represent is randomly-generated, i.e., their contents cannot coincide with that of other pieces of information), or *instance-random* (meaning that the information they represent is both randomly generated and unique for the given protocol instance, i.e., a nonce). Marking items as random or instance-random imposes restrictions (see Definition 5.1.3) on which instantiations we consider. For instance, if $\mathbf{k}|_l$ is marked random, then conclusion "$v|_l$ is derivable if key $\mathbf{k}|_l$ has the same contents as attribute $age|_l$" is not allowed because instantiations in which $\mathbf{k}|_l$ and $age|_l$ have the same contents are not allowed. These restrictions allow us to focus on interesting possibilities rather than instantiations that cannot happen in practice. Formally, all symbolic items form a *Symbolic Information Model*:

Definition 5.1.1. A *Symbolic Information Model* is a set $\mathfrak{P}^{\text{sym}}$ of *symbolic items* of the form $v|_a$; here, v is called the *variable*, and a is called the *profile*, such that:

- $\mathfrak{P}^{\text{sym}}$ is partitioned into *symbolic non-personal items* $\mathfrak{G}^{\text{sym}} \subset \mathfrak{P}^{\text{sym}}$ (with $a = \cdot$); *symbolic data items* $\mathfrak{D}^{\text{sym}} \subset \mathfrak{P}^{\text{sym}}$ (with $a \neq \cdot$); and *symbolic identifiers* $\mathfrak{I}^{\text{sym}} \subset \mathfrak{P}^{\text{sym}}$ (with $a \neq \cdot$).
- A symbolic variables v is either *non-random* (v is denoted non-boldfaced), *random* ($\mathbf{v}$ is denoted boldfaced), or *instance-random* ($\overline{\mathbf{v}}$ is denoted boldfaced and overlined)[1].

[1] Randomness is a property of the variable, independently from the profile. E.g., if v is random, then *all* symbolic items with variable v are random, i.e., of the form $\mathbf{v}|_k$

All protocols share the same Symbolic Information Model, but not all protocols use all symbolic items. Although the type (i.e., identifier/data item/non-personal information) and randomness (i.e. non-random/random/instance-random) of a symbolic item are defined orthogonally, not all combinations are meaningful. For instance, instance-random data should be regarded as non-personal as it does not reveal any information about a data subject.

Below we give an example of a Symbolic Information Model.

Example 5.1.2. Consider a protocol between a client and a server using the following information: the private key of the client; an identifier of a data subject; a data item representing the subject's age; and a fresh nonce. We model the client's private key as a symbolic identifier $\mathbf{k}^-|_{cli}$; because keys are assumed to be randomly-generated, it is random. The data subject's identifier and age are a non-random

symbolic identifier $id|_{su}$ and data item $age|_{su}$, respectively[2]. The nonce is an instance-random symbolic non-personal item $\overline{\mathbf{n}}|_.$; in particular, we assume that (the contents of) this nonce never re-occurs in any message outside this protocol instance. □

[2] Depending on the application, identifiers can be random (i.e., randomly generated) or not (e.g., sequentially generated). A data item representing an age is clearly not random.

Conclusions from reasoning about a Symbolic Information Model apply to any Information Model I that *instantiates* it. By this, we mean that I only contains context items $v|_k^\pi$ corresponding to symbolic items $v|_k/\mathbf{v}|_k/\overline{\mathbf{v}}|_k$ of the right type (identifier/data item/non-personal item); and that I respects randomness and instance-randomness. Formally:

Definition 5.1.3. Let $\mathfrak{P}^{\text{sym}}$ be a Symbolic Information Model, and $I = (\mathsf{P}^{\text{ctx}}, \mathcal{P}^{\text{inf}}, \mathbb{P}^{\text{cnt}}, \Leftrightarrow, \sigma, \tau)$ an Information Model. We say that I is an *instantiation* of $\mathfrak{P}^{\text{sym}}$ if:

- For each $d|_k^\pi \in \mathsf{P}^{\text{ctx}}$, corresponding symbolic item $d|_k/\mathbf{d}|_k/\overline{\mathbf{d}}|_k \in \mathfrak{P}^{\text{sym}}$ is of the same type (i.e., identifier/data item/non-personal item);
- If $\mathsf{p} = v|_k^\pi, \overline{\mathbf{v}}|_k$ is instance-random, and $\mathsf{p} \doteq \mathsf{p}'$, then $\mathsf{p} = \mathsf{p}'$;
- If $\mathsf{p} = v|_k^\pi, \mathbf{v}|_k$ is random, and $\mathsf{p} \doteq \mathsf{p}'$, then $\mathsf{p}' = v|_l^\eta$ for some η, l.

5

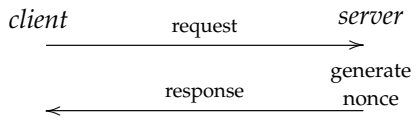
A random item with variable v can only be content equivalent to other (random) items with variable v. This simplifies our analysis; but to obtain accurate results, it is important to ensure that this assumption is reasonable. For instance, all private keys (from the same key space) should be modelled with the same random variable. Also, this definition technically states that all domains in the Information Model (including those that represent initial knowledge) are included in the symbolic model. In practice, we only model protocols symbolically, and leave the symbolic model of initial knowledge implicit.

We now demonstrate what conditions a Symbolic Information Model imposes on its instantiations.

Example 5.1.4. Consider the Symbolic Information Model $\mathfrak{P}^{\text{sym}} = \{\mathbf{k}^-|_{cli}, id|_{su}, age|_{su}, \overline{\mathbf{n}}|_.\}$ from Example 5.1.2. Suppose that Information Model $I = (\mathsf{P}^{\text{ctx}}, \mathcal{P}^{\text{inf}}, \mathbb{P}^{\text{cnt}}, \Leftrightarrow, \sigma, \tau)$ instantiates $\mathfrak{P}^{\text{sym}}$. In particular, suppose that contexts π and η model two protocol instances in which the same client and server exchange information about two different data subjects:

$$\{k^-|_{cli}^\pi, id|_{su}^\pi, age|_{su}^\pi, n|_.^\pi, k^-|_{cli}^\eta, id|_{su}^\eta, age|_{su}^\eta, n|_.^\eta\} \subset \mathsf{P}^{\text{ctx}}.$$

Because $\mathbf{k}^-|_{cli}$ is random, $k^-|_{cli}^\pi$ can be content equivalent to $k^-|_{cli}^\eta$ (which it will be, because the client is the same), but not to the other context items above. On the other hand, because $id|_{su}$ and $age|_{su}$ are not random, we could have $id|_{su}^\pi \doteq age|_{su}^\pi$, $id|_{su}^\pi \doteq age|_{su}^\eta$, $age|_{su}^\pi \doteq age|_{su}^\eta$, etcetera. (But not $id|_{su}^\pi \doteq id|_{su}^\eta$ because the data subjects of π and η are different.) Finally, because $\overline{\mathbf{n}}|_.$ is instance-random, we have that $n|_.^\pi \neq n|_.^\eta$; nor are other context items in P^{ctx} content equivalent to either $n|_.^\pi$ or $n|_.^\eta$. □



(a) Informal protocol description

$$P(cli) = \{\{\mathsf{pk}(\mathbf{k}^-|_{cli}), id|_{su}\}, \mathsf{aenc}(\{age|_{su}, \overline{\mathbf{n}}|.\}, \mathsf{pk}(\mathbf{k}^-|_{cli}))\}$$
$$P(srv) = \{\overline{\mathbf{n}}|., \{\mathsf{pk}(\mathbf{k}^-|_{cli}), id|_{su}\}, \mathsf{aenc}(\{age|_{su}, \overline{\mathbf{n}}|.\}, \mathsf{pk}(\mathbf{k}^-|_{cli}))\}$$

(b) Model as symbolic protocol

Figure 5.1: A simple protocol: informal description (left), model as a symbolic protocol (right)

A *symbolic protocol* specifies which messages are learned by each actor involved in a protocol instance. To model these messages, we use a signature Σ with associated set of construction and elimination rules as in Chapter 3. Messages built from symbolic items using function symbols of the signature are called *symbolic messages*. A *symbolic protocol* then captures, for each role in the protocol, what messages are sent, received, and generated by the actor performing that role. As before, neither the order of these messages matters, nor whether they were sent, received, or generated. Thus, we simply assign a set of symbolic messages to each role:

Definition 5.1.5. Let $\mathfrak{P}^{\text{sym}}$ be a Symbolic Information Model, and Σ a signature.

- The set $\mathfrak{L}^{sym}$ of *symbolic messages* is the set of formal terms built from symbolic items in $\mathfrak{P}^{\text{sym}}$ by recursive application $f(\mathfrak{m}_1, \ldots, \mathfrak{m}_k)$ of function symbols $f/k \in \Sigma$;
- A *symbolic protocol* P between roles $r_1, \ldots, r_k$ is a collection of sets $P(r_i) \subset \mathfrak{L}^{sym}$ of symbolic messages, where each set $P(r_i)$ contains all messages sent, received, and generated by the actor performing the role r_i in a full run of the protocol.

We now show a simple symbolic protocol.

Example 5.1.6. Continue with the Symbolic Information Model $\mathfrak{P}^{\text{sym}} = \{\mathbf{k}^-|_{cli}, id|_{su}, age|_{su}, \overline{\mathbf{n}}|.\}$ from Example 5.1.2. Consider a protocol between a client and a server, following the structure of Figure 5.1(a). First, the client sends a request to the server containing her public key and the identifier of a subject. The server generates a nonce, and responds with an asymmetric encryption with the client's public key of the subject's age and the nonce. We formalise this protocol as a symbolic protocol P between client cli and server srv, as shown in Figure 5.1(b). For the client, protocol role $P(srv)$ contains the messages he has sent and received. For the server, protocol role $P(srv)$ additionally contains the nonce he has generated. □

Intuitively, the knowledge base $\mathcal{C}_a$ of an actor a consists of *instantiations* of symbolic messages obtained through involvement in protocol instances. More precisely, the *instantiation* of symbolic message $\mathfrak{m}$ in domain π, denoted $\mathfrak{m}|^{\pi}$, is obtained by instantiating its symbolic items, e.g., $\mathsf{aenc}(\{age|_{su}, \overline{\mathbf{n}}|.\}, \mathsf{pk}(\mathbf{k}^-|_{cli}))|^{\pi} = \mathsf{aenc}(\{age|^{\pi}_{su}, n|^{\pi}_{\cdot}\}, \mathsf{pk}(k^-|^{\pi}_{cli}))$. Let $\mathfrak{C}|^{\pi} := \{\mathfrak{m}|^{\pi} \mid \mathfrak{m} \in \mathfrak{C}\}$. If domain π represents an instance of protocol P and actor a performs role r_i in that protocol instance, then this contributes set $\{\mathfrak{m}|^{\pi} \mid \mathfrak{m} \in P(r_i)\}$

of context messages to his knowledge base. Formally, we say that $(\mathcal{C}, \pi)$ is an *instantiation* of the set of symbolic messages $P(r_i)$:

Definition 5.1.7. Let $\mathfrak{C}$ be a set of symbolic messages, $\mathcal{C}$ a knowledge base, and π a domain. We say that $(\mathcal{C}, \pi)$ *instantiates* $\mathfrak{C}$ if $\mathfrak{C}|^{\pi} \subset \mathcal{C}$, and $\mathcal{C}$ contains no other messages containing context items with domain π.

Typically, a knowledge base $\mathcal{C}_a$ is the union $P_1(r_1)|^{\pi_1} \cup \ldots \cup P_k(r_k)|^{\pi_k}$ of such contributions. As discussed in Chapter 3, the knowledge base $\mathcal{C}_A$ of a coalition of actors is the union of the knowledge bases of the individual actors. In particular, if different actors in the coalition have performed different roles $r_1, \ldots, r_k$ in a single instance π of P, this contributes set $(P(r_1) \cup \ldots \cup P(r_k))|^{\pi}$ to the knowledge base of the coalition.

The next example demonstrates the relation between symbolic protocols and their instantiations.

Example 5.1.8. Consider the symbolic protocol P from Example 5.1.6, and knowledge base

$$\mathcal{C} = \{\{\mathsf{pk}(k^-|^{\pi}_{cli}), id|^{\pi}_{su}\}, \mathsf{aenc}(\{age|^{\pi}_{su}, n|^{\pi}_{.}\}, \mathsf{pk}(k^-|^{\pi}_{cli})),$$
$$n|^{\eta}_{.}, \{\mathsf{pk}(k^-|^{\eta}_{cli}), id|^{\eta}_{su}\}, \mathsf{aenc}(\{age|^{\eta}_{su}, n|^{\eta}_{.}\}, \mathsf{pk}(k^-|^{\eta}_{cli})).\}$$

Then $\mathcal{C}$ represents the knowledge of an actor who was involved as client in protocol instance π, and as server in protocol instance η. Formally, $(\mathcal{C}, \pi)$ instantiates $P(cli)$ and $(\mathcal{C}, \eta)$ instantiates $P(srv)$. □

5.2 Constraints

In the next two sections, we aim to capture what symbolic messages can be derived from a protocol instance under what conditions. We model these conditions by *constraints*. Constraints are boolean formulae with conjunction $\wedge$, disjunction $\vee$ and two types of atomic propositions: derivability and content equivalence constraints. *Derivability constraint* $\mathfrak{m}$ expresses that contents of message $\mathfrak{m}$ need to be known apart from the protocol instance. *Content equivalence constraint* $\mathfrak{m} \doteq \mathfrak{m}'$, expresses that messages $\mathfrak{m}, \mathfrak{m}'$ must have the same contents inside the protocol instance. **T** and **F** denote true and false:

Definition 5.2.1. Let γ be a constraint, $\mathcal{C}$ a knowledge base, and π a domain. Then γ is *satisfied* in $(\mathcal{C}, \pi)$ if: (i) $\gamma = \mathbf{T}$; (ii) $\gamma = \mathfrak{m}$, and $\mathcal{C} \vdash \mathfrak{m}'$ for some $\mathfrak{m}'$ outside of domain π[3] with $\mathfrak{m}|^{\pi} \doteq \mathfrak{m}'$; (iii) $\gamma = \mathfrak{m} \doteq \mathfrak{m}'$ and $\mathfrak{m}|^{\pi} \doteq \mathfrak{m}'|^{\pi}$; (iv) $\gamma = \gamma_1 \vee \gamma_2$ and γ_1 or γ_2 is satisfied; or (v) $\gamma = \gamma_1 \wedge \gamma_2$ and γ_1 and γ_2 are satisfied.

[3] I.e., $\mathfrak{m}'$ does not contain any context items with domain π;

Constraints can be manipulated as boolean formulae. Constraint γ_1 *implies* constraint γ_2 if, whenever γ_1 is satisfied in an instantiation, then γ_2 is also satisfied. Two constraints γ_1, γ_2 are called *equivalent* if they imply each other, i.e., γ_1 is satisfied in an instantiation exactly if γ_2 is. Constraint γ is called *trivial* if it is satisfied in any instantiation (otherwise *non-trivial*). Constraint γ is called *satisfiable*

if there exist instantiations in which it is satisfied (otherwise *non-satisfiable*). Hence, manipulation of a constraint as a boolean formula gives an equivalent constraint. For instance, replacing a trivial constraint by **T** or a non-satisfiable constraint by **F** gives an equivalent constraint.

The next example demonstrates the satisfaction of constraints as well as their manipulation as boolean formulae.

Example 5.2.2. Consider the Symbolic Information Model $\mathfrak{P}^{\text{sym}} = \{\mathbf{k}^-|_{cli}, id|_{su}, age|_{su}, \overline{\mathbf{n}}|.\}$ from Example 5.1.2. Then:

- Constraints $\gamma_1 = \mathbf{T}$ and $\gamma_2 = \mathbf{k}^-|_{cli} \doteq \mathbf{k}^-|_{cli}$ are trivial. As a consequence, γ_1 and γ_2 are equivalent.
- Constraint $\gamma_3 = \overline{\mathbf{n}}|.$ is non-satisfiable. Namely, because $\overline{\mathbf{n}}|.$ is instance-random, its contents cannot be known apart from the protocol instance it occurs in. Formally, by Definition 5.1.3, for any $(\mathcal{C}, \pi)$ there is no $\mathfrak{m}'$ with domain other than π such that $n|^{\pi}_{\cdot} \doteq \mathfrak{m}'$. As a consequence, γ_3 is equivalent to **F**.
- Constraint $\gamma_4 = \mathbf{k}^-|_{cli} \wedge (id|_{su} \vee id|_{su} \doteq age|_{su})$ is non-trivial and satisfiable. By boolean logic, it is equivalent to $\gamma_5 = (\mathbf{k}^-|_{cli} \wedge id|_{su}) \vee (\mathbf{k}^-|_{cli} \wedge id|_{su} \doteq age|_{su})$. Moreover, it implies $\gamma_6 = \mathbf{k}^-|_{cli}$.

Consider now the knowledge base $\mathcal{C}$ from Example 5.1.8. Because γ_1 and γ_2 are trivial, they are satisfied in $(\mathcal{C}, \pi)$ and $(\mathcal{C}, \eta)$. Because γ_3 is non-satisfiable, it is not satisfied. In this case, γ_4 is also not satisfied in $(\mathcal{C}, \pi)$ and $(\mathcal{C}, \eta)$. Namely, for γ_4 to be satisfied in $(\mathcal{C}, \pi)$ (resp. $(\mathcal{C}, \eta)$), a message content equivalent to $k^-|^{\pi}_{cli}$ (resp. $k^-|^{\eta}_{cli}$) needs to be derivable: one verifies that this is not the case. Also, γ_5 and γ_6 are not satisfied. □

5.3 Symbolic Derivability

We now introduce the symbolic derivability relation, that captures exactly what messages can be derived from an instantiation of a symbolic protocol under what conditions. This relation generalises the $\vdash^-$ relation introduced in Section 3.6. Recall that, to derive a message using $\vdash^-$, no elimination rules are applied to the result of a construction rule; and that, under the EC and CC assumptions from Section 3.6, this is sufficient to derive an actor's view.[4] Hence, our objective is to find out, given a set $\mathfrak{C}$ of symbolic messages and a symbolic message $\mathfrak{m}$, which constraints are needed for $\mathcal{C} \vdash^- \mathfrak{m}|^{\pi}$ to hold. More precisely, we are looking for constraints that are *necessary* (if the message can be derived, then the constraint holds) and *sufficient* (if the constraint holds, then the message can be derived):

Definition 5.3.1. Let $\mathfrak{C}$ be a set of symbolic messages, $\mathfrak{m}$ a symbolic message, and γ a constraint. Then: (i) γ is *sufficient* for $\mathfrak{m}$ in $\mathfrak{C}$ if whenever $(\mathcal{C}, \pi)$ instantiates $\mathfrak{C}$ and γ is satisfied in $(\mathcal{C}, \pi)$, then $\mathcal{C} \vdash \mathfrak{m}|^{\pi}$; (ii) γ is *necessary* for $\mathfrak{m}$ if $\mathfrak{C}$ whenever $(\mathcal{C}, \pi)$ instantiates $\mathfrak{C}$ and $\mathcal{C} \vdash \mathfrak{m}|^{\pi}$, then γ is satisfied in $(\mathcal{C}, \pi)$.

[4] Intuitively, generalising $\vdash^-$ rather than $\vdash$ is easier because it allows for a cleaner separation between "internal" messages $\mathfrak{m}|^{\pi}$ and external messages. For instance, it can no longer happen that message $\mathfrak{m}|^{\pi}$ is derived by elimination from a message that both contains context items from π and from other contexts because such a message would need to be constructed first. As a consequence, the structure of auxiliary messages needed to derive $\mathfrak{m}|^{\pi}$ only depends on messages in $\mathfrak{C}$. Generalising $\vdash$ does not seem straightforward.

If there are satisfiable sufficient constraints for $\mathfrak{m}$ in $\mathfrak{C}$, then we call $\mathfrak{m}$ *possibly derivable* from $\mathfrak{C}$. If γ is both necessary and sufficient for $\mathfrak{m}$ in $\mathfrak{C}$, then we simply call γ a constraint for $\mathfrak{m}$ in $\mathfrak{C}$.

We demonstrate necessity and sufficiency by a small example:

Example 5.3.2. Consider the symbolic protocol role

$$\mathfrak{C} = \{\{\mathsf{pk}(\mathbf{k}^-|_{cli}), id|_{su}\}, \mathsf{aenc}(\{age|_{su}, \overline{\mathbf{n}}|.\}, \mathsf{pk}(\mathbf{k}^-|_{cli}))\}$$

from Example 5.1.6. Suppose that an actor knows messages instantiating $\mathfrak{C}$, say in domain π. Clearly, if he knows the contents of key $k^-|^{\pi}_{cli}$, he can obtain $age|^{\pi}_{su}$ by decrypting the asymmetric encryption. Hence, $\gamma_1 = \mathbf{k}^-|_{cli}$ is a sufficient constraint for $age|_{su}$. (As a consequence, $age|_{su}$ is possibly derivable.) Using this secret key is also the only way to obtain $age|^{\pi}_{su}$, i.e., γ_1 is also necessary. Similarly, $\gamma_2 = \mathbf{k}^-|_{cli} \wedge (id|_{su} \vee id|_{su} \doteq age|_{su})$ is sufficient but not necessary, and $\gamma_3 = id|_{su} \vee id|_{su} \doteq age|_{su}$ is neither necessary nor sufficient. □

We find sufficient constraints by generalising the deductive system for $\vdash^-$ from Chapter 3. Namely, we present a deductive system that defines the notation $\mathfrak{C} \vdash^s \gamma \Rightarrow \mathfrak{m}$, where $\mathfrak{C}$ is a set of symbolic messages; γ is a constraint; and $\mathfrak{m}$ is a symbolic message. Intuitively, we ensure that our deductive system is "sound" (i.e., any constraint γ is sufficient) and "complete" (i.e., whenever an instantiation of $\mathfrak{m}$ can be derived, the deductive system gives a satisfied constraint γ). As a consequence, the disjunction $\gamma_1 \vee \ldots \vee \gamma_k$ of all γ_i such that $\mathfrak{C} \vdash^s \gamma_i \Rightarrow \mathfrak{m}$ is necessary for $\mathfrak{m}$.[5]

[5] Indeed, our deductive system will ensure that there are only finitely many such γ_i.

To define the deductive system, we first specify how construction and elimination rules act on symbolic messages. Analogously to the instantiated case, a *symbolic instantiation* of construction rule $\mathsf{f}(a_1, \ldots, a_k) \leftarrow b_1, \ldots, b_l$ is a substitution σ of symbolic messages for all variables in the rule, in which case we write

$$\mathsf{f}(a_1\sigma, \ldots, a_k\sigma) \leftarrow b_1\sigma, \ldots, b_l\sigma.$$

A *symbolic instantiation* of elimination rule $\mathsf{f}(a_1, \ldots, a_k) \xrightarrow{\doteq b_1, \ldots, \doteq b_l} c$ is a substitution σ of symbolic messages for all variables in the rule, in which case we write $\mathsf{f}(a_1\sigma, \ldots, a_k\sigma) \xrightarrow{\doteq b_1\sigma, \ldots, \doteq b_l\sigma} c\sigma$[6].

[6] Unlike in the instantiated case, we apply σ directly to the b_i. Our deductive system ensures that also messages content equivalent to $b_i\sigma$ are allowed in derivations.

The idea of our symbolic deductive system (Figure 5.2) is to symbolically mimic all operations of the instantiated deductive system (Figure 3.4). Namely, suppose that $\mathcal{C} \vdash^- \mathfrak{m}|^{\pi}$, where $(\mathcal{C}, \pi)$ instantiates $\mathfrak{C}$. Then $\mathfrak{m}|^{\pi}$ is obtained by first applying rule $(\vdash^- \mathbf{0})$ using a message in the knowledge base, and then repeatedly applying rule $(\vdash^- \mathbf{E})$. We mimic these steps by symbolic rules $(\vdash^{\mathbf{s}} \mathbf{0})$ and $(\vdash^{\mathbf{s}} \mathbf{E})$. When applying $(\vdash^{\mathbf{s}} \mathbf{E})$, auxiliary messages are used that are obtained by possibly applying rule $(\vdash^+ \mathbf{C})$ to applications of rule $(\vdash^+ \mathbf{0})$. We mimic $(\vdash^+ \mathbf{C})$ by $(\vdash^{\mathbf{c}} \mathbf{C})$. auxiliary messages only need to be known up to content equivalence; hence we mimic $(\vdash^+ \mathbf{0})$ by allowing message $\mathfrak{m}$ itself to be derivable from $\mathfrak{C}$ $(\vdash^{\mathbf{c}} \mathbf{0})$; or a message $\mathfrak{m}'$ to be derivable from $\mathfrak{C}$ for which $\mathfrak{m} \doteq \mathfrak{m}'$ is is satisfiable $(\vdash^{\mathbf{c}} \mathbf{1})$; or a message from outside the protocol instance to be derivable that is content equivalent to $\mathfrak{m}$ $(\vdash^{\mathbf{c}} \mathbf{2})$.

$$\frac{}{\mathfrak{C}\vdash^{s}\mathbf{T}\Rightarrow\mathfrak{m}}\,(\mathfrak{m}\in\mathfrak{C})\ (\vdash^{\mathbf{s}}\mathbf{0}) \qquad \frac{\mathfrak{C}\vdash^{s}\gamma\Rightarrow\mathfrak{m}\quad\mathfrak{C}\vdash^{c}\gamma_1\Rightarrow\mathfrak{n}_1\quad\dots\quad\mathfrak{C}\vdash^{c}\gamma_k\Rightarrow\mathfrak{n}_k}{\mathfrak{C}\vdash^{s}\gamma\wedge\gamma_1\wedge\ldots\wedge\gamma_k\Rightarrow\mathfrak{n}}\,(\mathfrak{m}\xrightarrow{\doteq\mathfrak{n}_1,\ldots,\doteq\mathfrak{n}_k}\mathfrak{n})\ (\vdash^{\mathbf{s}}\mathbf{E})$$

$$\frac{\mathfrak{C}\vdash^{s}\gamma\Rightarrow\mathfrak{m}}{\mathfrak{C}\vdash^{c}\gamma\Rightarrow\mathfrak{m}}\,(\vdash^{\mathbf{c}}\mathbf{0}) \qquad \frac{\mathfrak{C}\vdash^{s}\gamma\Rightarrow\mathfrak{m}}{\mathfrak{C}\vdash^{c}\gamma\wedge\mathfrak{m}\doteq\mathfrak{m}'\Rightarrow\mathfrak{m}'}\,(\mathfrak{m}\doteq\mathfrak{m}'\text{ satisfiable})\ (\vdash^{\mathbf{c}}\mathbf{1}) \qquad \frac{}{\mathfrak{C}\vdash^{c}\mathfrak{m}\Rightarrow\mathfrak{m}}\,(\mathfrak{m}\text{ satisfiable})\ (\vdash^{\mathbf{c}}\mathbf{2})$$

$$\frac{\mathfrak{C}\vdash^{c}\gamma_1\Rightarrow\mathfrak{n}_1\quad\dots\quad\mathfrak{C}\vdash^{c}\gamma_l\Rightarrow\mathfrak{n}_l}{\mathfrak{C}\vdash^{c}\gamma_1\wedge\ldots\wedge\gamma_l\Rightarrow\mathsf{f}(\mathfrak{m}_1,\ldots,\mathfrak{m}_k)}\,(\mathsf{f}(\mathfrak{m}_1,\ldots,\mathfrak{m}_k)\leftarrow\mathfrak{n}_1,\ldots,\mathfrak{n}_l)\ (\vdash^{\mathbf{c}}\mathbf{C})$$

Figure 5.2: Inference rules for the symbolic derivability relation ($\mathfrak{C}$ a set of symbolic messages; γ, γ_i constraints; $\mathfrak{m}$, $\mathfrak{m}_i$, $\mathfrak{n}$, $\mathfrak{n}_i$ symbolic messages). In any conjunction $\gamma_1\wedge\ldots\wedge\gamma_k$, duplicate and trivial constraints are left out

We obtain the following definition and proposition:

Definition 5.3.3. Let $\mathfrak{C}$ be a set of symbolic messages, $\mathfrak{m}$ a symbolic message, and γ a constraint. We say that $\mathfrak{m}$ is *symbolically derivable* from $\mathfrak{C}$ using γ, denoted $\mathfrak{C}\vdash^{s}\gamma\Rightarrow\mathfrak{m}$, if the conclusion $\mathfrak{C}\vdash^{s}\gamma\Rightarrow\mathfrak{m}$ follows from the deductive system of Figure 5.2.

Proposition 5.3.4. Let $\mathfrak{C}$ be a set of symbolic messages, $\mathcal{C}$ be a knowledge base and π a domain such that $(\mathcal{C},\pi)$ instantiates $\mathfrak{C}$. Let $\mathfrak{m}$ be a symbolic message. Then:

$$\mathcal{C}\vdash^{-}\mathfrak{m}|^{\pi}\ \text{ iff }\ \exists\gamma:\mathfrak{C}\vdash^{s}\gamma\Rightarrow\mathfrak{m}\wedge\gamma\text{ satisfied in }(\mathcal{C},\pi).$$

Proof. (Sketch.) If $\mathcal{C}\vdash^{-}\mathfrak{m}|^{\pi}$, then consider the derivation in tree form[7]. Replace all largest subtrees that derive messages that have no context items from domain π by applications of $(\vdash^{\mathbf{s}}\mathbf{2})$. Then, replace applications of $(\vdash^{-}\mathbf{0})$ by $(\vdash^{\mathbf{s}}\mathbf{0})$; applications of $(\vdash^{+}\mathbf{0})$ by $(\vdash^{\mathbf{c}}\mathbf{0})$ or $(\vdash^{\mathbf{c}}\mathbf{1})$ depending on their later use in $(\vdash^{+}\mathbf{C})$; $(\vdash^{-}\mathbf{E})$ by $(\vdash^{\mathbf{s}}\mathbf{E})$ and $(\vdash^{+}\mathbf{C})$ by $(\vdash^{\mathbf{c}}\mathbf{C})$. The result is a proof tree of $\mathfrak{C}\vdash^{s}\gamma\Rightarrow\mathfrak{m}$ so that γ is satisfied in $(\mathcal{C},\pi)$. This proves the "only if" part of the Proposition.

For the "if" part of the Proposition, suppose $\mathfrak{C}\vdash^{s}\gamma\Rightarrow\mathfrak{m}$ and perform the translation in the opposite direction. In particular, for $(\vdash^{\mathbf{c}}\mathbf{2})$, $\mathfrak{m}$ is satisfied so there is a proof tree for $\mathcal{C}\vdash^{-}\mathsf{m}$ for some m with $\mathsf{m}\doteq\mathfrak{m}|^{\pi}$, so replace the application of $(\vdash^{\mathbf{c}}\mathbf{2})$ by this proof tree. The result is valid a proof tree for $\mathcal{C}\vdash^{-}\mathfrak{m}|^{\pi}$. □

[7] I.e., in which each node represents a valid application of an inference rule

We now present some example symbolic derivations.

Example 5.3.5. Consider the symbolic protocol role

$$\mathfrak{C}=\{\{\mathsf{pk}(\mathbf{k}^{-}|_{cli}),id|_{su}\},\mathsf{aenc}(\{age|_{su},\overline{\mathbf{n}}|.\},\mathsf{pk}(\mathbf{k}^{-}|_{cli}))\}$$

from Example 5.1.6. In Example 5.3.2, we claimed that $\mathbf{k}^{-}|_{cli}$ is sufficient for $age|_{su}$. Indeed, we have $\mathfrak{C}\vdash^{s}\mathbf{k}^{-}|_{cli}\Rightarrow age|_{su}$ by the derivation

$$\dfrac{\dfrac{\dfrac{}{\mathfrak{C}\vdash^{s}\mathbf{T}\Rightarrow\mathsf{aenc}(\{age|_{su},\overline{\mathbf{n}}|.\},\mathsf{pk}(\mathbf{k}^{-}|_{cli}))}\,(\vdash^{\mathbf{s}}\mathbf{0}) \quad \dfrac{}{\mathfrak{C}\vdash^{c}\mathbf{k}^{-}|_{cli}\Rightarrow\mathbf{k}^{-}|_{cli}}\,(\vdash^{\mathbf{c}}\mathbf{2})}{\mathfrak{C}\vdash^{s}\mathbf{k}^{-}|_{cli}\Rightarrow\{age|_{su},\overline{\mathbf{n}}|.\}}\,(\vdash^{\mathbf{s}}\mathbf{E})}{\mathfrak{C}\vdash^{s}\mathbf{k}^{-}|_{cli}\Rightarrow age|_{su}}\,(\vdash^{\mathbf{s}}\mathbf{E})$$

Figure 5.3: Symbolic derivation from Example 5.3.5

$$\dfrac{\dfrac{}{\mathfrak{C}\vdash^{s}\mathbf{T}\Rightarrow\mathsf{aenc}(d_1|_u,\mathsf{pk}(\mathbf{k}^-|_{srv}))}(\vdash^{\mathbf{s}}\mathbf{0}) \quad \dfrac{\dfrac{}{\mathfrak{C}\vdash^{s}\mathbf{T}\Rightarrow d_2|_u}(\vdash^{\mathbf{s}}\mathbf{0})}{\mathfrak{C}\vdash^{c} d_2|_u\doteq d_1|_u\Rightarrow d_1|_u}(\vdash^{\mathbf{c}}\mathbf{1}) \quad \dfrac{\dfrac{}{\mathfrak{C}\vdash^{s}\mathbf{T}\Rightarrow\mathsf{pk}(\mathbf{k}^-|_{srv})}(\vdash^{\mathbf{s}}\mathbf{0})}{\mathfrak{C}\vdash^{c}\mathbf{T}\Rightarrow\mathsf{pk}(\mathbf{k}^-|_{srv})}(\vdash^{\mathbf{c}}\mathbf{0})}{\mathfrak{C}\vdash^{s} d_2|_u\doteq d_1|_u\Rightarrow d_1|_u}(\vdash^{\mathbf{s}}\mathbf{E})$$

Figure 5.4: Symbolic derivation from Example 5.3.6

in Figure 5.3. As mentioned, if γ_i are all the constraints such that $\mathfrak{C}\vdash^{s}\gamma_i\Rightarrow\mathfrak{age}|_{su}$, then $\gamma_1\vee\ldots\vee\gamma_k$ is necessary. In this example, all such constraints γ_i imply $\mathbf{k}^-|_{cli}$[8], so $\mathbf{k}^-|_{cli}$ is necessary and sufficient for $age|_{su}$, i.e., it is a constraint for $age|_{su}$ in $\mathfrak{C}$. □

[8] For instance, $\mathbf{k}^-|_{cli}\wedge\mathsf{pk}(\mathbf{k}^-|_{cli})$ can be derived by applying a reconstruction rule to obtain $\mathsf{pk}(\mathbf{k}^-|_{cli})$ from the encryption. This constraint clearly implies $\mathbf{k}^-|_{cli}$.

Example 5.3.6. Consider the set

$$\mathfrak{C}=\{\mathsf{aenc}(d_1|_u,\mathsf{pk}(\mathbf{k}^-|_{srv})),\mathsf{pk}(\mathbf{k}^-|_{srv}),d_2|_u\}$$

of symbolic messages. The constraint $d_1|_u\vee d_2|_u\doteq d_1|_u\vee\mathbf{k}^-|_{srv}$ is necessary and sufficient for $d_1|_u$. For instance, $\mathfrak{C}\vdash^{s} d_2|_u\doteq d_1|_u\Rightarrow d_1|_u$ follows by the derivation shown in Figure 5.4. □

5

5.4 Equatability

We now extend the above approach to direct equatability (Definition 3.5.2), the principle behind detectability and associability (Definition 3.5.3).

We first extend our notion of constraints to reason about two domains at once. For direct equatability, an actor needs to derive a message m_1 in one domain; it needs to be content equivalent to a message m_2 in the second domain; and he needs to derive the second message. We capture these conditions about two different domains with *biconstraints*. As before, they are boolean formulae with conjunction $\wedge$ and disjunction $\vee$, but now with different atomic propositions. Namely, we have derivability constraints of the form $(\mathfrak{m})_l$, $(\mathfrak{m})_r$, depending on whether they apply to the first or second domain; and similarly for content equivalence constraints $(\mathfrak{m}\doteq\mathfrak{m}')_l$, $(\mathfrak{m}\doteq\mathfrak{m}')_r$. In addition, we introduce *correspondence constraints* $(\mathfrak{m})_l\doteq(\mathfrak{m}')_r$, meaning that $\mathfrak{m}$ and $\mathfrak{m}'$ should be content equivalent in their respective domains. Satisfaction of biconstraints is defined as follows.

Definition 5.4.1. Let γ be a biconstraint, $\mathcal{C}$ a knowledge base, and π,κ two distinct domains. Then γ is *satisfied* in $(\mathcal{C},\pi)$, $(\mathcal{C},\kappa)$ if one of the following holds:

- $\gamma=\mathbf{T}$;
- $\gamma=(\gamma')_l$, and γ' is satisfied in $(\mathcal{C},\pi)$; or $\gamma=(\gamma')_r$, and γ' is satisfied in $(\mathcal{C},\kappa)$;
- $\gamma=(\mathfrak{m})_l\doteq(\mathfrak{m}')_r$, and $\mathfrak{m}|^{\pi}\doteq\mathfrak{m}'|^{\kappa}$;

- $\gamma = \gamma_1 \vee \gamma_2$ and γ_1 or γ_2 is satisfied; or $\gamma = \gamma_1 \wedge \gamma_2$ and γ_1, γ_2 are both satisfied.

Given a constraint γ, the biconstraints $(\gamma)_l$, $(\gamma)_r$ are defined in the obvious way, e.g., $(\gamma_1 \vee \gamma_2)_l = (\gamma_1)_l \vee (\gamma_2)_l$. Note that γ is satisfied in $(\mathcal{C}, \pi)$ iff $(\gamma)_l$ is satisfied in $(\mathcal{C}, \pi)$, $(\mathcal{C}, \kappa)$ (for any κ); and similarly for $(\gamma)_r$. Implication, equivalence, triviality, and satisfiability of biconstraints are defined analogously to above.

We briefly demonstrate the above concepts:

Example 5.4.2. Consider the Symbolic Information Model

$$\mathfrak{P}^{\mathrm{sym}} = \{\mathbf{k}^-|_{cli}, id|_{su}, age|_{su}, \overline{\mathbf{n}}|.\}$$

from Example 5.1.2.

Biconstraint $\gamma_1 = (\mathbf{k}^-|_{cli})_l \doteq (\mathbf{k}^-|_{cli})_r$ is satisfiable. Namely, it is satisfied in $(\mathcal{C}, \pi)$, $(\mathcal{C}, \eta)$ if the same secret key is used in both domains π, η. In particular, because $\mathbf{k}^-|_{cli}$ is a symbolic identifier, this means that the client in both contexts must be the same. On the other hand, biconstraint $\gamma_2 = (\overline{\mathbf{n}}|.)_l \doteq (\overline{\mathbf{n}}|.)_r$ is not satisfiable because $\overline{\mathbf{n}}|.$ is instance-random.

Consider now knowledge base

$$\begin{aligned}\mathcal{C} = \{\{\mathsf{pk}(k^-|^{\pi}_{cli}), id|^{\pi}_{su}\}, \mathsf{aenc}(\{age|^{\pi}_{su}, n|^{\pi}_{\cdot}\}, \mathsf{pk}(k^-|^{\pi}_{cli})),\\ n|^{\eta}_{\cdot}, \{\mathsf{pk}(k^-|^{\eta}_{cli}), id|^{\eta}_{su}\}, \mathsf{aenc}(\{age|^{\eta}_{su}, n|^{\eta}_{\cdot}\}, \mathsf{pk}(k^-|^{\eta}_{cli}))\}\end{aligned}$$

from Example 5.1.8, where $k^-|^{\pi}_{cli} \doteq k^-|^{\eta}_{cli}$. Then γ_1 is satisfied in $(\mathcal{C}, \pi)$, $(\mathcal{C}, \eta)$. Also, $\gamma_3 = (\mathsf{pk}(\mathbf{k}^-|_{cli}))_l \vee (id|_{su})_r$ is satisfied. Namely, message $\mathsf{pk}(k^-|^{\eta}_{cli})$ content equivalent to $\mathsf{pk}(k^-|^{\pi}_{cli})$ can be derived, so $(\mathsf{pk}(\mathbf{k}^-|_{cli}))_l$ is satisfied. Finally, $\gamma_4 = (id|_{su})_r$ is satisfied if and only if $id|^{\pi}_{su} \doteq id|^{\eta}_{su}$. Namely, $id|^{\pi}_{su}$ is the only derivable message outside domain η that could possibly be content equivalent to $id|^{\eta}_{su}$. □

We now define when constraints are necessary or sufficient to directly equate symbolic items $\mathfrak{p}, \mathfrak{p}'$. There are two cases: either $\mathfrak{p}$ and $\mathfrak{p}'$ are both instantiated in the same domain, or not. In the former case, the constraints for $\mathfrak{p} \doteq \mathfrak{p}'$ are constraints, and we call them internally sufficient or internally necessary. In the latter case, the constraints are biconstraints, and we call them sufficient or necessary.

Definition 5.4.3. Let $\mathfrak{C}, \mathfrak{C}'$ be sets of symbolic messages; $\mathfrak{p}, \mathfrak{p}'$ symbolic items. Let γ be a constraint, and γ' a biconstraint.

- γ is *internally sufficient* for $\mathfrak{p} \doteq_0 \mathfrak{p}'$ in $\mathfrak{C}$ if, whenever γ is satisfied in instantiation $(\mathcal{C}, \pi)$ of $\mathfrak{C}$, then $\mathcal{C} \vdash \mathfrak{p}|^{\pi} \doteq_0 \mathfrak{p}'|^{\pi}$;
- γ' is *sufficient* for $\mathfrak{p} \doteq_0 \mathfrak{p}'$ in $\mathfrak{C}, \mathfrak{C}'$ if, whenever γ is satisfied in instantiations $(\mathcal{C}, \pi)$, $(\mathcal{C}, \kappa)$ of $\mathfrak{C}, \mathfrak{C}'$ (where $\pi \neq \kappa$), then $\mathcal{C} \vdash \mathfrak{p}|^{\pi} \doteq_0 \mathfrak{p}'|^{\kappa}$;
- γ is *internally necessary* for $\mathfrak{p} \doteq_0 \mathfrak{p}'$ in $\mathfrak{C}$ if whenever $\mathcal{C} \vdash \mathfrak{p}|^{\pi} \doteq_0 \mathfrak{p}'|^{\pi}$ in instantiation $(\mathcal{C}, \pi)$ of $\mathfrak{C}$, then γ' is satisfied.

- γ' is *necessary* for $\mathfrak{p} \doteq_0 \mathfrak{p}'$ in $\mathfrak{C}, \mathfrak{C}'$ if whenever $\mathcal{C} \vdash\!\cdots \mathfrak{p}|^{\pi} \doteq_0 \mathfrak{p}'|^{\kappa}$ in instantiations $(\mathcal{C}, \pi), (\mathcal{C}, \kappa)$ of $\mathfrak{C}, \mathfrak{C}'$ (where $\pi \neq \kappa$), then γ' is satisfied.

If γ' is necessary and sufficient for $\mathfrak{p} \doteq_0 \mathfrak{p}'$ in $\mathfrak{C}, \mathfrak{C}'$, then we simply call γ' a biconstraint for $\mathfrak{p} \doteq_0 \mathfrak{p}'$ in $\mathfrak{C}, \mathfrak{C}'$; if γ is internally sufficient and internally necessary for $\mathfrak{p} \doteq_0 \mathfrak{p}'$ in $\mathfrak{C}$ then we call it a constraint for $\mathfrak{p} \doteq_0 \mathfrak{p}'$ in $\mathfrak{C}$.

Note that, if $\mathfrak{C} = \mathfrak{C}'$, then biconstraints γ' for $\mathfrak{p} \doteq_0 \mathfrak{p}'$ in $\mathfrak{C}, \mathfrak{C}'$ deal with the case when $\mathfrak{p}, \mathfrak{p}'$ come from *different instances* of $\mathfrak{C}$. On the other hand, constraints γ for $\mathfrak{p} \doteq_0 \mathfrak{p}'$ in $\mathfrak{C}$ deal with the case when $\mathfrak{p}, \mathfrak{p}'$ are from the same instance (i.e., domain).

We demonstrate the definition by continuing our running example.

Example 5.4.4. Recall the symbolic protocol roles from Example 5.1.6:

$$\mathfrak{C}_1 = \{\{\mathsf{pk}(\mathbf{k}^-|_{cli}), id|_{su}\}, \mathsf{aenc}(\{age|_{su}, \overline{\mathbf{n}}|.\}, \mathsf{pk}(\mathbf{k}^-|_{cli}))\};$$
$$\mathfrak{C}_2 = \{\overline{\mathbf{n}}|., \{\mathsf{pk}(\mathbf{k}^-|_{cli}), id|_{su}\}, \mathsf{aenc}(\{age|_{su}, \overline{\mathbf{n}}|.\}, \mathsf{pk}(\mathbf{k}^-|_{cli}))\}\}.$$

We claim that $\gamma_1 = (\mathbf{k}^-|_{cli})_l \doteq (\mathbf{k}^-|_{cli})_r$ is a biconstraint for $\mathbf{k}^-|_{cli} \doteq_0 \mathbf{k}^-|_{cli}$ in $\mathfrak{C}_1, \mathfrak{C}_2$. That is, whenever an actor has performed the roles corresponding to $\mathfrak{C}_1$ and $\mathfrak{C}_2$, he knows if the client's private key was the same or not. Indeed, both symbolic protocol roles contain the client's public key from which this knowledge follows directly.

Also, $\gamma_2 = (id|_{su})_l \doteq (age|_{su})_r$ is a biconstraint for $id|_{su} \doteq_0 age|_{su}$ in $\mathfrak{C}_1, \mathfrak{C}_2$. Indeed, it is clearly necessary. However, it is also sufficient: if γ_2 holds in instantiations $(\mathcal{C}, \pi), (\mathcal{C}, \kappa)$ of $\mathfrak{C}_1, \mathfrak{C}_2$, then clearly $\mathcal{C} \vdash\!\cdots^{-} id|^{\pi}_{su}$. Also, we have derivability of $\mathsf{pk}(k^-|^{\kappa}_{cli}), n|^{\kappa}_{\cdot}$, and a message content equivalent to $age|^{\kappa}_{su}$, hence $age|^{\kappa}_{su}$ can be derived using the reconstruction rule for aenc. So, whenever γ_2 is satisfied, $id|^{\pi}_{su}$ and $age|^{\kappa}_{su}$ are derivable, hence directly equatable, i.e., γ_1 is sufficient.

On the other hand, $\gamma_2 = (age|_{su})_l \doteq (id|_{su})_r$ is not sufficient for $age|_{su} \doteq_0 id|_{su}$ in $\mathfrak{C}_1, \mathfrak{C}_2$. Namely, if γ_2 holds in instantiations $(\mathcal{C}, \pi)$, $(\mathcal{C}, \kappa)$ of $\mathfrak{C}_1, \mathfrak{C}_2$, then $\mathcal{C} \vdash\!\cdots age|^{\pi}_{su}$ does not necessarily hold: the actor does not know $n|^{\pi}_{\cdot}$ and hence needs $k^-|^{\pi}_{cli}$ to derive $age|^{\pi}_{su}$. Indeed, $\gamma_3 = (\mathbf{k}^-|_{cli})_l \wedge (age|_{su})_l \doteq (id|_{su})_r$ is necessary and sufficient. □

We now show how to find biconstraints for direct equatability. Let $\mathfrak{C}$ and $\mathfrak{C}'$ be two symbolic protocol roles, and $\mathfrak{p}, \mathfrak{p}'$ two symbolic items for which we want to analyse direct equatability. Intuitively, we need to find messages from the two protocols that may be content equivalent, may be derivable by the actor, and from which content equivalence of the respective items can be concluded. Formally, we find all $(\gamma_i, \mathfrak{m}_i, \gamma'_i, \mathfrak{m}'_i)$ such that:

- for some z, $\mathfrak{m}_i@z = \mathfrak{p}$ and $\mathfrak{m}'_i@z = \mathfrak{p}'$;
- $\gamma_i \neq \mathbf{F}$ is a constraint for $\mathfrak{m}_i$ in $\mathfrak{C}$;
- $\gamma'_i \neq \mathbf{F}$ is a constraint for $\mathfrak{m}'_i$ in $\mathfrak{C}'$;
- $\mathfrak{m}_i \doteq \mathfrak{m}'_i$ is satisfiable in $\mathfrak{C}, \mathfrak{C}'$.

It follows directly from Proposition 3.6.5 that the is a biconstraint for $\mathfrak{p} \doteq_0 \mathfrak{p}'$:

$$\bigvee_i \left((\gamma_i)_l \wedge (\mathfrak{m}_i \doteq \mathfrak{m}'_i) \wedge (\gamma'_i)_r \right).$$

We demonstrate the procedure with the following example.

Example 5.4.5. We continue with Example 5.4.4. We use the above procedure to find biconstraints for $age|_{su} \doteq_0 id|_{su}$ in $\mathfrak{C}_1, \mathfrak{C}_2$. For this, we need to find tuples $(\gamma_i, \mathfrak{m}_i, \gamma'_i, \mathfrak{m}'_i)$ satisfying the above conditions.

First, let us find possible candidates for γ'_i and $\mathfrak{m}'_i$: that is, symbolic messages $\mathfrak{m}'_i$ such that $\mathfrak{m}'_i@z = id|_{su}$ for some z, and $\gamma'_i \neq \mathbf{F}$ is a constraint for $\mathfrak{m}_i$ in $\mathfrak{C}_2$. Clearly, the only two possibilities are $(\mathbf{T}, \{\mathsf{pk}(\mathbf{k}^-|_{cli}), id|_{su}\})$ (with $z = 2$) and $(\mathbf{T}, id|_{su})$ (with $z = \epsilon$).

First, consider candidate $(\mathbf{T}, \{\mathsf{pk}(\mathbf{k}^-|_{cli}), id|_{su}\})$. We need to find $\gamma_i, \mathfrak{m}_i$ such that $\mathfrak{m}_i@2 = age|_{su}$, and $\gamma_i \neq \mathbf{F}$ is a constraint for $\mathfrak{m}_i$. Moreover, $(\mathfrak{m}_i)_l \doteq (\{\mathsf{pk}(\mathbf{k}^-|_{cli}), id|_{su}\})_r$ needs to be satisfiable. Clearly, such a pair does not exist: there is no $\mathfrak{m}_i$ with $age|_{su}$ at the required location.

Now, consider candidate $(\mathbf{T}, id|_{su})$. We now need to find $\gamma_i, \mathfrak{m}_i$ such that $\mathfrak{m}_i@\epsilon = age|_{su}$; $\gamma_i \neq \mathbf{F}$ is a constraint for $\mathfrak{m}_i$; and $(\mathfrak{m}_i)_l \doteq (id|_{su})_r$ is satisfiable. In this case, $\mathfrak{m}_i@\epsilon = age|_{su}$ means that $\mathfrak{m}_i = age|_{su}$; moreover, one checks that $\gamma_i = \mathbf{k}^-|_{cli}$ is a constraint for $age|_{su}$ in $\mathfrak{C}_1$.

Since this is the only possible candidate, we conclude that

$$(\mathbf{T})_l \wedge (age|_{su})_l \doteq (id|_{su})_r \wedge (\mathbf{k}^-|_{cli})_r$$

is a biconstraint for $age|_{su} \doteq_0 id|_{su}$ in $\mathfrak{C}_1, \mathfrak{C}_2$. Indeed, this biconstraint is equivalent to the one we found in Example 5.4.4. □

5.5 Analysing Detectability and Associability: Constraint Graph

In this section, we introduce the *constraint graph*, a visualisation of relevant constraints for detectability and associability. Let $\{P_i(r_i)|_{p_i}\}$ be a set of *symbolic profiles*: profiles p_i in protocol roles $P_i(r_i)$ whose privacy we are interested in. The constraint graph shows under what constraints pieces of personal information from these profiles can be derived or equated to information from other protocols (leading to detectability); and under what constraints these profiles can be associated to each other. We give a decision procedure showing how privacy guarantees can be derived from this graph.

We now introduce a running example we will use throughout this section.

Example 5.5.1. Consider an information system with three "authentication protocols" P_i, $i = 1, 2, 3$. In each variant i, service provider sp receives identifier $id|_u$ and data $data|_u$ about user u. In variant P_1, the identifier and data are sent directly. In variant P_2, the identifier and data are encrypted for a trusted third party ttp using its public key $\mathsf{pk}(\mathbf{k}^-|_{ttp})$. In variant P_3, a nonce $\overline{\mathbf{n}}|_.$ is added to ensure the

Constraint	Message	CE
$\top$	$id\|_u$	1
$\top$	$data\|_u$	1

(a) Derivation table for $P_1(sp)$

Constraint	Message	CE
$\top$	$\mathsf{aenc}(\{id\|_u, data\|_u\}, \mathsf{pk}(\mathbf{k}^-\|_{ttp}))$	2
$(\mathsf{pk}(\mathbf{k}^-\|_{ttp}) \wedge data\|_u \wedge id\|_u) \vee \mathbf{k}^-\|_{ttp}$	$id\|_u$	1
$(\mathsf{pk}(\mathbf{k}^-\|_{ttp}) \wedge data\|_u \wedge id\|_u) \vee \mathbf{k}^-\|_{ttp}$	$data\|_u$	1
$(\mathsf{pk}(\mathbf{k}^-\|_{ttp}) \wedge data\|_u \wedge id\|_u) \vee \mathbf{k}^-\|_{ttp}$	$\mathsf{pk}(\mathbf{k}^-\|_{ttp})$	3
$\mathbf{k}^-\|_{ttp}$	$\mathbf{k}^-\|_{ttp}$	4

(b) Derivation table for $P_2(sp)$

Constraint	Message	CE
$\top$	$\mathsf{aenc}(\{id\|_u, data\|_u, \overline{\mathbf{n}}\|.\}, \mathsf{pk}(\mathbf{k}^-\|_{ttp}))$	5
$\mathbf{k}^-\|_{ttp}$	$id\|_u$	1
$\mathbf{k}^-\|_{ttp}$	$data\|_u$	1
$\mathbf{k}^-\|_{ttp}$	$\overline{\mathbf{n}}\|.$	6
$\mathbf{k}^-\|_{ttp}$	$\mathbf{k}^-\|_{ttp}$	4
$\mathbf{k}^-\|_{ttp}$	$\mathsf{pk}(\mathbf{k}^-\|_{ttp})$	3

(c) Derivation table for $P_3(sp)$

Figure 5.5: Derivation tables for authentication protocols (Example 5.5.1)

encryption is different every time:

$$P_1(sp) = \{\{id|_u, data|_u\}\}$$
$$P_2(sp) = \{\mathsf{aenc}(\{id|_u, data|_u\}, \mathsf{pk}(\mathbf{k}^-|_{ttp}))\}$$
$$P_3(sp) = \{\mathsf{aenc}(\{id|_u, data|_u, \overline{\mathbf{n}}|.\}, \mathsf{pk}(\mathbf{k}^-|_{ttp}))\}$$

In this section, we will analyse the privacy of the user in these three authentication protocols. Hence, we are interested in the symbolic profiles $P_1(sp)|_u$, $P_2(sp)|_u$, and $P_3(sp)|_u$. □

As a first step towards building the constraint graph, we determine *derivation tables* showing constraints and possible content equivalences for each symbolic message. Namely, for each protocol role $P_i(r_i)$ considered, we list possibly derivable messages $\mathfrak{m}$ (except lists[9]) and their constraints γ. We divide these messages $\mathfrak{m}$ from all protocol roles into equivalence sets, numbered 1, 2, ..., according to possible content equivalence. Namely, $\mathfrak{m}_a$ and $\mathfrak{m}_b$ are in the same equivalence set iff $(\mathfrak{m}_a)_l \doteq (\mathfrak{m}_b)_r$ is satisfiable, either as a constraint (i.e., within one protocol instance), or as a biconstraint (i.e., between two protocol instances). For instance, the derivation tables for the symbolic protocols from Example 5.5.1 are shown in Figure 5.5.

[9] Lists are irrelevant for our purposes because any conclusions drawn from a list can also be drawn from its elements

The *constraint graph* for a given set $\mathfrak{P}$ of symbolic profiles consists of two types of nodes: *profile nodes* and *content equivalence nodes*, and labelled, undirected edges between them. Profile nodes are labelled by a symbolic profile, and visualised as a box containing entries $\gamma \Rightarrow d|_p$, where $d|_p$ is a piece of personal information from the profile and γ is its constraint. Intuitively, they show what information from the profile can be derived. Content equivalence nodes are labelled n@z, where n represents one of the above sets of equivalent messages, and z is a position in the messages. Edges between profile nodes and content equivalence nodes are labelled by $\gamma \Rightarrow d|_p$, where γ is a constraint and $d|_p$ is a symbolic item; and are drawn solid or dashed, as explained below. Intuitively, content equivalence nodes

represent messages that are potentially useful for equating pieces of personal information; edges indicate relevance of these messages to the symbolic profile they are connected to.

The constraint graph is built by performing the following steps for each symbolic profile $P(r)|_p \in \mathfrak{P}$:

- Add a profile node for $P(r)|_p \in \mathfrak{P}$, containing entries $\gamma \Rightarrow d|_p$ for all items with profile p that are possibly derivable from $P(r)$;
- For each message $\mathfrak{m}$ with constraint $\gamma \neq \mathbf{F}$ in $P(r)$ (with label $\boxed{\mathrm{n}}$), and z such that $\mathfrak{m}@z$ is a symbolic identifier $d|_p$, add content equivalence node $\boxed{\mathrm{n}}@z$ (if it does not exist), and solid edge between $P(r)|_p$ and $\boxed{\mathrm{n}}@z$ labelled $\gamma \Rightarrow d|_p$;
- For each message $\mathfrak{m}$ with constraint $\gamma \neq \mathbf{F}$ in $P(r)$ (with label $\boxed{\mathrm{n}}$), and $z \neq \epsilon$ such that $\mathfrak{m}@z$ is a symbolic data item $d|_p$, add content equivalence node $\boxed{\mathrm{n}}@z$ (if it does not exist), and dashed edge between $P(r)|_p$ and $\boxed{\mathrm{n}}@z$ labelled $\gamma \Rightarrow d|_p$;

Intuitively, derivability follows from the constraints shown inside a profile node; equatability follows from links between profile nodes and content equivalence nodes. For symbolic identifiers, equatability is relevant both for detectability and for associability; in this case, we draw solid edges between the profile node and all $\boxed{\mathrm{n}}@z$ representing the identifier. For symbolic data items, equatability is relevant only for detectability; in this case, we do not allow $z = \epsilon$[10] and draw a dashed edge. Constraint graphs can be simplified in two ways. First, if the messages $\mathfrak{m}_i$ represented by $\boxed{\mathrm{n}}$ contain instance-random symbolic items and all $\mathfrak{m}_i@z$ are the same, the node and edges can be removed. Second, edges $\gamma_1 \Rightarrow d|_p, \ldots, \gamma_k \Rightarrow d|_p$ between the same nodes can be combined into one edge $\gamma_1 \vee \gamma_k \Rightarrow d|_p$.

We now apply the above procedure in our running example:

Example 5.5.2. Consider the symbolic profiles

$$\mathfrak{P} = \{P_1(sp)|_u, P_2(sp)|_u, P_3(sp)|_u\}$$

from Example 5.5.1. We construct the constraint graph for $\mathfrak{P}$ according to the above procedure. The result is shown in Figure 5.6.

For $P_1(sp)|_u$, we add a profile node containing $\mathbf{T} \Rightarrow id|_u$, $\mathbf{T} \Rightarrow data|_u$. We also add content equivalence node $\boxed{1}@\epsilon$ and solid edge $\mathbf{T} \Rightarrow id|_u$, representing that the $\boxed{1}$-labelled message $id|_u$ may be used to associate $P_1(sp)|_u$ to other profiles. Note that we do not have edge

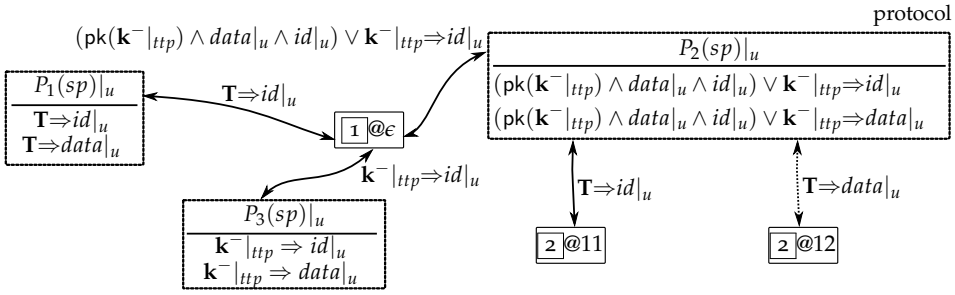


Figure 5.6: Constraint graph for three variants of a simple authentication protocol (Example 5.5.3)

[10] If $z = \epsilon$, i.e., the data item itself is used for equatability, then the data item can also be derived directly without equatability

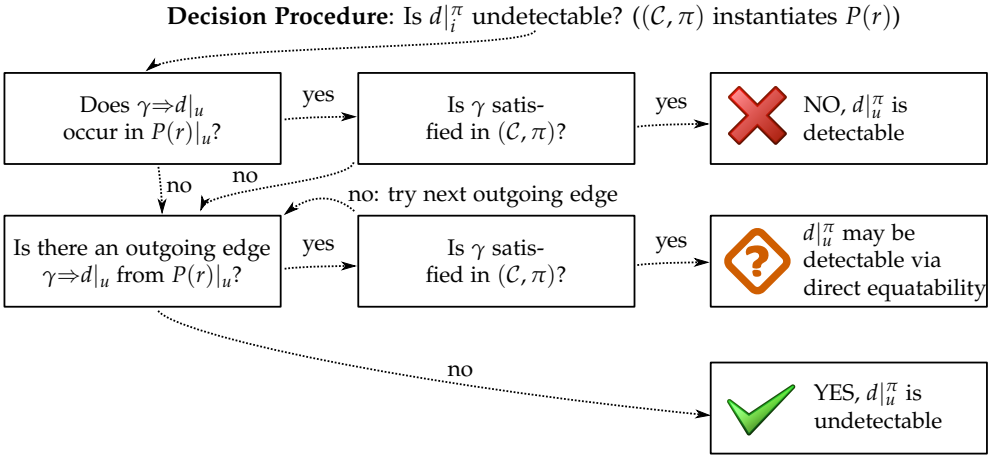


Figure 5.7: Decision procedure for undetectability of a context item using the constraint graph

$\mathbf{T} \Rightarrow data|_u$ because the condition that the position $\neq \epsilon$ is not satisfied; intuitively, if we can derive $data|_u$ then we don't need equatability to detect it.

For $P_2(sp)|_u$, we also add an edge connecting it to $\boxed{1}@\epsilon$. In addition, we add content equivalence nodes $\boxed{2}@11$, $\boxed{2}@12$ representing possible equatability of $id|_u$ and $data|_u$ from message $\mathsf{aenc}(\{id|_u, data|_u\}, \mathsf{pk}(\mathbf{k}^-|_{ttp}))$.

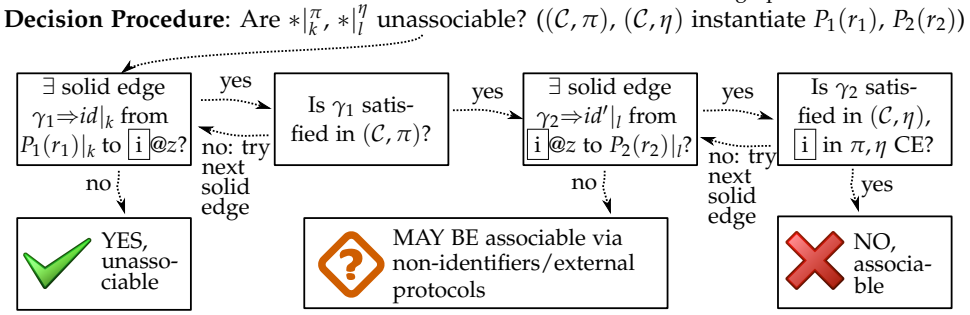
For $P_3(sp)|_u$, we do not need to add content equivalence nodes $\boxed{5}@11$, $\boxed{5}@12$ because $\mathsf{aenc}(\{id|_u, data|_u, \overline{\mathbf{n}}|_.\}, \mathsf{pk}(\mathbf{k}^-|_{ttp}))$ contains an instance-random item and is the only $\boxed{5}$-labelled message. □

In Figure 5.7, we present a decision procedure for undetectability using a constraint graph. To guarantee undetectability of an instantiation $d|_u^\pi$ of symbolic item $d|_u$, we need to guarantee that the item can neither be derived directly, nor equated to another detectable item. Hence, we first check if $d|_u^\pi$ can be derived by checking its constraints γ mentioned in the profile node (top line of boxes in the figure). Then, we check if any messages $\mathfrak{m}_i|^\pi$ exist that may be used to equate $d|_u^\pi$ to another context item, and whose constraint γ_i is satisfied (middle line of boxes). If this is the case, then equatability may be possible, so further study is needed.[11] On the other hand, if neither $d|_u^\pi$ nor a message containing it has satisfied constraints, we can guarantee undetectability of $d|_u^\pi$ (bottom box). In summary, γ is sufficient for detectability of $d|_u$, and $\gamma \vee \bigvee_i(\gamma_i)$ is necessary.

In Figure 5.8, we present a decision procedure for unassociability

[11] Namely, we can use the derivation table to see which messages from considered protocols may be content equivalent to $\mathfrak{m}_i|^\pi$, and hence to which context items from considered protocols $d|_u^\pi$ may be equated. However, messages content equivalent to $\mathfrak{m}|^\pi$ may also occur in protocols that are not in the constraint graph, so we cannot guarantee undetectability.

Figure 5.8: Decision procedure for unassociability using the constraint graph



using a constraint graph. To guarantee unassociability of instantiations $*|_k^\pi$, $*|_l^\eta$ of symbolic profiles $P_1(r_1)|_k$, $P_2(r_2)|_l$, we need to guarantee that no identifiers from the two instantiations can be directly equated, and that no indirect associability via other contexts is possible. For simplicity, our decision procedure only checks for direct equatability and otherwise returns inconclusively. Namely, we check if two content equivalent messages of the form $\boxed{\text{i}}$ exist in $P_1(r_1)$, $P_2(r_2)$, such that one message contains identifier $id|_k$ and has satisfied constraints γ_1 in $P_1(r_1)|_k$, and the other message contains identifier $id'|_l$ and has satisfied constraints γ_2 in $P_2(r_2)|_l$ (top line of boxes in the figure). If this is the case, then $*|_k^\pi$ and $*|_l^\eta$ are associable. If no such message $\boxed{\text{i}}$ exists for $P_1(r_1)|_k$, then $*|_k^\pi$ and $*|_l^\eta$ are not associable. In other cases, further study is needed[12].

[12] Namely, if there is an outgoing solid edge between $P_1(r_1)$ and $\boxed{\text{i}}@z$, we can use the derivation tables to see what context items in other protocol instances the respective identifier may be equated to. However, messages content equivalent to $\boxed{\text{i}}$ may also occur in protocols that are not in the constraint graph, so we cannot guarantee unassociability.

The following example demonstrates constraint graphs and the privacy guarantees that can be derived from them:

Example 5.5.3. Consider the constraint graph from Figure 5.6. Detectability of $data|_u$ for the different protocols can be analysed by looking at the profile nodes, and their outgoing edges with label $\gamma \Rightarrow data|_u$ (cf. Figure 5.7). In P_1, $data|_u$ has constraint $\mathbf{T}$, hence is always detectable. In P_2, $data|_u$ can be derived by either decrypting or reconstructing the encryption; it may also be detectable via equatability if encryption $\boxed{2}$ occurs in other instances of P_2 or outside of the system. In P_3, because of nonce $\overline{\mathbf{n}}|_\cdot$, $data|_u$ can be detected only if $\mathbf{k}^-|_{ttp}$ is known.

Associability of the user in different protocol instances is analysed using the solid edges between profile nodes and content equivalence nodes (cf. Figure 5.8). The $\mathbf{T} \Rightarrow id|_u$-labelled edge between $P_1(sp)|_u$ and $\boxed{1}@\epsilon$ indicates that the user profile in an instance of P_1 can be associated to any other context in which $id|_u$ can be derived. The $\mathbf{k}^-|_{ttp} \Rightarrow id|_u$-labelled edge between $P_3(sp)|_u$ and $\boxed{1}\epsilon$ shows that the user in P_3 can only be linked to other contexts if $\mathbf{k}^-|_{ttp}$ is known. The edge between $P_2(sp)|_u$ and $\boxed{2}@11$ shows that users from instances of P_2 with the same identifier, data and TTP are associable: this is because of the deterministic encryption. Moreover, the edge between $P_2(sp)|_u$ and $\boxed{1}@\epsilon$ shows that the user in P_2 may be associable to the user in P_1 or P_3 if constraint $\gamma = (\mathsf{pk}(\mathbf{k}^-|_{ttp}) \wedge data|_u \wedge id|_u) \vee \mathbf{k}^-|_{ttp}$ is satisfied, i.e., if the identifier in P_2 can be derived. The constraint graph guarantees that, if γ is not satisfied and message $\boxed{2}$ does not occur in other protocol instances, then the user profile in P_2 is not associable to other contexts. □

5.6 Implementation

In this section, we present an algorithm for computing constraints using the symbolic derivability relation introduced in Section 5.3. The algorithm presented in this section works under the same assumptions as the instantiated algorithm in Section 3.7, namely, the EC and EE assumptions (that guarantee correctness of the deduct-

Algorithm 2 Given $\mathfrak{C}$, compute all pairs $(\gamma, \mathfrak{m})$ s.t. γ is constraint for $\mathfrak{m}$

{first, compute symbolic derivability without testing rules}
$\mathfrak{C}^{new} := \{(\mathbf{T}, \mathfrak{m}) \mid \mathfrak{m} \in \mathfrak{C}\}$
repeat
  $\mathfrak{C}^{old} := \mathfrak{C}^{new}$
  $\mathfrak{C}^{+}, \mathfrak{C}^{new} := \varnothing$
  {try to apply all non-testing elimination rules}
  for all $(\gamma_0, \mathfrak{m}) \in \mathfrak{C}^{old}$, inst. $\mathfrak{m} \xrightarrow{\doteq \mathfrak{m}_1, \ldots, \doteq \mathfrak{m}_k} \mathfrak{n}$ of non-testing rules **do**
    for $i = 1$ to k **do**
      find all $\gamma_{i,j}$ such that $\mathfrak{C}^{old} \vdash^{c*} \gamma_{i,j} \Rightarrow \mathfrak{m}_i$
      $\gamma_i := (\bigvee_j \gamma_{i,j})$ {collect all constraints for $\mathfrak{m}_i$}
    end for
    $\mathfrak{C}^{+} := \mathfrak{C}^{+} \cup \{(\gamma_0 \wedge \gamma_1 \wedge \ldots \wedge \gamma_k, \mathfrak{n})\}$
  end for
  {for each message, collect all constraints found}
  $\mathfrak{M} := \{\mathfrak{m} \mid \exists \gamma \not\equiv \mathbf{F} : (\gamma, \mathfrak{m}) \in \mathcal{C}^{old} \cup \mathfrak{C}^{+}\}$ {messages with satisfiable constr.}
  for all $\mathfrak{m} \in \mathfrak{M}$ **do**
    $\gamma := \{\gamma' \mid (\gamma', \mathfrak{m}) \in \mathfrak{C}^{old} \cup \mathfrak{C}^{+}\}$
    $\mathfrak{C}^{new} := \mathfrak{C}^{new} \cup \{(\vee\gamma, \mathfrak{m})\}$
  end for
until $\mathfrak{C}^{old} \equiv \mathfrak{C}^{new}$ {i.e., all constraints equivalent as predicates}
{compute all derivable messages using precomputed $\mathfrak{C}^{new}$}
$\mathfrak{C}^{ret} := \varnothing$
for all possibly derivable $\mathfrak{m}$ **do**
  find all γ_j such that $(\mathfrak{C}, \mathfrak{C}^{new}) \vdash^{s*} \gamma_j \Rightarrow \mathfrak{m}$
  $\mathfrak{C}^{ret} = \mathfrak{C}^{ret} \cup \{(\bigvee_j \gamma_j, \mathfrak{m})\}$ {$\bigvee_j \gamma_j$ is necessary and sufficient for $\mathfrak{m}$}
end for
return $\mathfrak{C}^{ret}$

ive system presented in this chapter, see Proposition 5.3.4), and the finiteness assumption (that guarantees termination). This algorithm has been implemented in a Prolog-based tool for the symbolic analysis of privacy in communication protocols[13].

[13] Available at `http://code.google.com/p/objective-privacy/`

Our algorithm (Algorithm 2) is based on the two same basic ideas as the instantiated algorithm in Section 3.7. Namely, the first idea is to determine derivable messages by induction on the number of recursive applications of $(\vdash^{\mathbf{s}} \mathbf{E})$. The present algorithm differs from the instantiated one because each iteration does not just add new messages, but also new constraints for existing messages. Indeed, at the kth step, we determine which constraints γ_k are sufficient to derive a message $\mathfrak{m}$ using $\leq k$ recursive applications of $(\vdash^{\mathbf{s}} \mathbf{E})$. We keep a set $\mathfrak{C}^k$ of such tuples $(\gamma_k, \mathfrak{m})$, and write $\mathfrak{C}^k \vdash^{c*} \gamma \Rightarrow \mathfrak{m}$ if $\mathfrak{C} \vdash^{c} \gamma \Rightarrow \mathfrak{m}$ holds with $\leq k$ recursive applications of $(\vdash^{\mathbf{s}} \mathbf{E})$. Relation $\vdash^{c*}$ follows from $\mathfrak{C}^k$ by the inference rules in Figure 5.9. We then use $\vdash^{c*}$ to see which elimination rules can be applied in the next iteration. If, for all messages $\mathfrak{m}$, the constraints from step k and from step $k+1$ are the same[14], then the found constraints must be necessary as well as sufficient.

[14] I.e., if they are equivalent as predicates

The second idea is that $\vdash^{c}$ can be evaluated without the use of testing rules. More precisely, one can show that for every deriva-

tion $\mathfrak{C}\vdash^{c} \gamma\Rightarrow\mathfrak{m}$, there exists a derivation $\mathfrak{C}\vdash^{c} \gamma'\Rightarrow\mathfrak{m}$ that does not use testing rules such that γ implies γ' (as constraints). Hence, we first iteratively determine $\mathfrak{C}^1,\ldots,\mathfrak{C}^k$ as above, but without the use of testing rules. Having computed $\mathfrak{C}^k$, we use it to determine constraints for message $\mathfrak{m}$ by evaluating $\vdash^{s}$ using $\vdash^{c*}$; formally, we evaluate the $\vdash^{s*}$ relation as defined in Figure 5.9.

Our algorithm terminates because the main **repeat**...**until** loop is executed finitely many times. Namely, first, note that only finitely many messages are possibly derivable, and that constraints are predicates over a finite set of symbols. Now, for each message $\mathfrak{m}$ with constraint γ, count the number of assignments of true and false to these symbols such that γ does *not* hold. Because each iteration replaces γ with a constraint γ such that $\gamma \Rightarrow \gamma'$, for each message this number is decreasing. Moreover, we only continue iterating if, for at least one message $\mathfrak{m}$, we have that $\gamma' \not\Rightarrow \gamma$, Hence, for that message, the number is *strictly* decreasing. This implies termination.

5.7 *Variable-Length Lists*

In this section, we outline how the above model can be extended to deal with variable-length lists. The above model of symbolic protocols assumes that the messages in all instantiations of the protocol have the exact same form. This means that we cannot capture protocols in which the amount of information differs between protocol instances. For instance, we cannot capture a protocol that sends an encrypted list of all attributes about a user. Variable-length lists address exactly this limitation.

The idea is introduce a new kind of symbolic message called a *var-list* (extending Definition 5.1.5). A var-list is a message $\{\mathfrak{m}\}_F$, where $\mathfrak{m}$ is a symbolic message, and F is a *family* capturing the kind of list ("all user attributes", "all revealed attributes", ...). Intuitively, $\{\mathfrak{m}\}_F$ represents an arbitrary number of messages of the form $\mathfrak{m}$. In an instantiation, $\{\mathfrak{m}\}_F$ is replaced by a list $\{\mathfrak{m}@_F1,\ldots,\mathfrak{m}@_Fk\}$ of actual copies of $\mathfrak{m}$ (where k depends on the instance). Here, $\mathtt{f}(\mathfrak{m}_1,\ldots,\mathfrak{m}_n)@_Fj = \mathtt{f}(\mathfrak{m}_1@_Fj,\ldots,\mathfrak{m}_n@_Fj)$[15] and $v|_p@_Fj = v@_Fj|_v$. Symbolic items $v@_Fj|_v$ are called *var-items* (extending Definition 5.1.1). Var-items $v@_Fj|_v$ are instantiated to context items $v@_Fj|_v^\pi$ as usual; Definition 5.1.3 is extended so that var-items behave the same as symbolic items with the same variable. For instance, if variable $\mathbf{v}$

[15] If a var-list occurs in a list $\mathfrak{m}$, then we merge the lists. E.g., we instantiate $\{id|_u, \{d|_u\}_{all}\}$ by $\{id|_u, d@_{all}1|_u, d@_{all}2|_u\}$ instead of $\{id|_u, \{d@_{all}1|_u, d@_{all}2|_u\}\}$.

Figure 5.9: Inference rules for the modified symbolic derivability relation ($\mathfrak{C}$ a set of pairs $(\gamma,\mathfrak{m})$, with γ a constraint and $\mathfrak{m}$ a symbolic message; γ, γ_i constraints; $\mathfrak{m}$, $\mathfrak{m}_i$, $\mathfrak{n}$, $\mathfrak{n}_i$ symbolic messages). In any conjunction $\gamma_1\wedge\ldots\wedge\gamma_k$, duplicate and $\mathbf{T}$ constraints are left out

$$\frac{}{\mathfrak{C}\vdash^{c*}\gamma\Rightarrow\mathfrak{m}}\,((\gamma,\mathfrak{m})\in\mathfrak{C})\ (\vdash^{\mathbf{c*}}\mathbf{0}) \qquad \frac{}{\mathfrak{C}\vdash^{c*}\gamma\wedge\mathfrak{m}\doteq\mathfrak{m}'\Rightarrow\mathfrak{m}}\,\frac{(\mathfrak{m}\doteq\mathfrak{m}'\text{ sat.},}{(\gamma,\mathfrak{m}')\in\mathfrak{C})}\ (\vdash^{\mathbf{c*}}\mathbf{1}) \qquad \frac{}{\mathfrak{C}\vdash^{c*}\mathfrak{m}\Rightarrow\mathfrak{m}}\,(\mathfrak{m}\text{ sat.})\ (\vdash^{\mathbf{c*}}\mathbf{2})$$

$$\frac{\mathfrak{C}\vdash^{c*}\gamma_1\Rightarrow\mathfrak{n}_1 \quad\ldots\quad \mathfrak{C}\vdash^{c*}\gamma_l\Rightarrow\mathfrak{n}_l}{\mathfrak{C}\vdash^{c*}\gamma_1\wedge\ldots\wedge\gamma_l\Rightarrow\mathtt{f}(\mathfrak{m}_1,\ldots,\mathfrak{m}_k)}\,(\mathtt{f}(\mathfrak{m}_1,\ldots,\mathfrak{m}_k)\leftarrow\mathfrak{n}_1,\ldots,\mathfrak{n}_l)\ (\vdash^{\mathbf{c*}}\mathbf{C})$$

$$\frac{}{(\mathfrak{C},\mathfrak{C}^k)\vdash^{s*}\mathbf{T}\Rightarrow\mathfrak{m}}\,(\mathfrak{m}\in\mathfrak{C})\ (\vdash^{\mathbf{s*}}\mathbf{0}) \qquad \frac{(\mathfrak{C},\mathfrak{C}^k)\vdash^{s*}\gamma\Rightarrow\mathfrak{m} \quad \mathfrak{C}^k\vdash^{c*}\gamma_1\Rightarrow\mathfrak{n}_1 \quad\ldots\quad \mathfrak{C}^k\vdash^{c*}\gamma_k\Rightarrow\mathfrak{n}_k}{(\mathfrak{C},\mathfrak{C}^k)\vdash^{s*}\gamma\wedge\gamma_1\wedge\ldots\wedge\gamma_k\Rightarrow\mathfrak{n}}\,(\mathfrak{m}\xrightarrow{\doteq\mathfrak{n}_1,\ldots,\doteq\mathfrak{n}_k}\mathfrak{n})\ (\vdash^{\mathbf{s*}}\mathbf{E})$$

is random, then instantiation $v@_{all}1|^{\pi}_{u}$ of a var-item can be content equivalent to any $v|^{\eta}_{l}$ or $v@_{F}k|^{\mu}_{n}$.

We now show how var-lists can be used to model more general protocols.

Example 5.7.1. Consider a variation of the symbolic protocol from Example 5.1.6. Suppose that, instead of the age of the user, we send a list of all of the user's attributes. This is modelled by the following symbolic message (where family *all* represent "all attributes"):

$$\mathsf{aenc}(\{\{d|_{su}\}_{all}, \overline{\mathbf{n}}|.\}, \mathsf{pk}(\mathbf{k}^{-}|_{cli})).$$

For instance, this message can be instantiated by an encryption of two data items:

$$\mathsf{aenc}(\{d@_{all}1|^{\pi}_{su}, d@_{all}2|^{\pi}_{su}, \overline{\mathbf{n}}|.^{\pi}\}, \mathsf{pk}(\mathbf{k}^{-}|^{\pi}_{cli})).$$

Instead of modelling an encryption of a list of data items, we can also model a list of encryptions of single data items:

$$\{\mathsf{aenc}(\{d|_{su}, \overline{\mathbf{n}}|.\}, \mathsf{pk}(\mathbf{k}^{-}|_{cli}))\}_{all}.$$

For instance, this message can be instantiated by:

$$\{\mathsf{aenc}(\{d@_{all}1|^{\pi}_{su}, \overline{\mathbf{n}}@_{all}1|.^{\pi}\}, \mathsf{pk}(\mathbf{k}^{-}@_{all}1|^{\pi}_{cli})),$$
$$\mathsf{aenc}(\{d@_{all}2|^{\pi}_{su}, \overline{\mathbf{n}}@_{all}2|.^{\pi}\}, \mathsf{pk}(\mathbf{k}^{-}@_{all}2|^{\pi}_{cli}))\}.$$

Note that, by instance-randomness, $\overline{\mathbf{n}}@_{all}1|.$ and $\overline{\mathbf{n}}@_{all}2|.$ are two *different* nonces, i.e., in any instantiation, $\overline{\mathbf{n}}@_{all}1|.^{\pi} \neq \overline{\mathbf{n}}@_{all}2|.^{\pi}$. □

The definitions of symbolic (bi)constraints and their necessity and sufficiency remain unchanged; however, constraints that are both necessary and sufficient may no longer exist. We demonstrate this in an example.

Example 5.7.2. Consider the following symbolic protocol role

$$\mathfrak{C} = \{d|_{u}, \{\mathsf{enc}(e|_{u}, f|_{u})\}_{all}\}.$$

For any j, k, the following constraints are satisfiable:

$$d|_{u} \doteq e@_{all}j|_{u} \qquad d|_{u} \doteq f@_{all}j|_{u} \qquad e@_{all}j|_{u} \doteq f@_{all}j|_{u}.$$

Moreover, for any j, $f@_{all}j|_{u}$ and $d|_{u} \doteq f@_{all}j|_{u}$ are sufficient constraints for $e@_{all}j|_{u}$. The first constraint shows that, for any index j, we can use the respective decryption key $f@_{all}j|_{u}$ to obtain the respective plaintext $e@_{all}j|_{u}$. The second constraint shows that, for any index j for which $d|_{u}$ happens to have the same contents as $f@_{all}j|_{u}$, item $e@_{all}j|_{u}$ can be derived.

However, there are infinitely many more sufficient constraints for $e@_{all}j|_{u}$. Namely, for any k, the following constraint is sufficient:

$$d|_{u} \doteq f@_{all}k|_{u} \wedge e@_{all}k|_{u} \doteq f@_{all}j|_{u}.$$

Indeed, one plaintext in the var-list may be the decryption key for another encryption. Because k can be arbitrarily high, there is no single necessary constraint for $e@_{all}j|_{u}$ other than $\mathbf{T}$; in particular, no constraint can be both necessary and sufficient. □

Figure 5.10: Derivations using pattern constraints (Example 5.7.3)

$$\dfrac{\dfrac{\dfrac{}{\mathfrak{C}\vdash^{s}\mathbf{T}\Rightarrow\{\mathsf{enc}(e|_u, f|_u)\}_{all}}(\vdash^{\mathbf{s}}\mathbf{0})}{\mathfrak{C}\vdash^{s}\mathbf{T}\Rightarrow\mathsf{enc}(e@_{all}?|_u, f@_{all}?|_u)}(\vdash^{\mathbf{s}}@) \quad \dfrac{}{\mathfrak{C}\vdash^{c} f@_{all}?|_u\Rightarrow f@_{all}?|_u}(\vdash^{\mathbf{c}}\mathbf{2})}{\mathfrak{C}\vdash^{s} f@_{all}?|_u\Rightarrow e@_{all}?|_u}(\vdash^{\mathbf{s}}\mathbf{E})$$

(a) Deriving plaintext with key from list

$$\dfrac{\dfrac{\dfrac{}{\mathfrak{C}\vdash^{s}\mathbf{T}\Rightarrow\{\mathsf{enc}(e|_u, f|_u)\}_{all}}(\vdash^{\mathbf{s}}\mathbf{0})}{\mathfrak{C}\vdash^{s}\mathbf{T}\Rightarrow\mathsf{enc}(e@_{all}?|_u, f@_{all}?|_u)}(\vdash^{\mathbf{s}}@) \quad \dfrac{\dfrac{}{\mathfrak{C}\vdash^{s}\mathbf{T}\Rightarrow d|_u}(\vdash^{\mathbf{s}}\mathbf{0})}{\mathfrak{C}\vdash^{s} d|_u\doteq f@_{all}?|_u\Rightarrow f@_{all}?|_u}(\vdash^{\mathbf{c}}\mathbf{1})}{\mathfrak{C}\vdash^{s} d|_u\doteq f@_{all}?|_u\Rightarrow e@_{all}?|_u}(\vdash^{\mathbf{s}}\mathbf{E})$$

(b) Deriving plaintext using other protocol message

To deal with the possibility of having infinitely many constraints, we reason about var-lists by abstracting away from particular indices. Namely, we introduce *pattern constraints* and *pattern messages* in which we allow symbolic items $f@_{all}?|_u$, where ? stands for "any index". We add the following rule to the deductive system for symbolic derivability (Figure 5.2):

$$\dfrac{\mathfrak{C}\vdash^{s}\gamma\Rightarrow\{\mathfrak{m}\}_F}{\mathfrak{C}\vdash^{s}\gamma\Rightarrow\mathfrak{m}@_F?}(\vdash^{\mathbf{s}}@)$$

Here, $\mathfrak{m}@_F?$ is defined like $\mathfrak{m}@_F k$, e.g., $d|_u@_{all}? = d@_{all}?|_u$. Intuitively, rule $(\vdash^{\mathbf{s}}@)$ states that, from a var-list, any item can be derived. The deductive system with this rule is evaluated as the original one. Because of the approximation introduced, the necessity and sufficiency guarantees of Proposition 5.3.4 no longer apply. However, we conjecture that derived constraints are necessary "up to patterns". That is, if $\mathcal{C}\vdash^{-}\mathfrak{m}|^{\pi}$, then there exist pattern constraint γ' and pattern message $\mathfrak{m}'$ such that $\mathfrak{C}\vdash^{s}\gamma'\Rightarrow\mathfrak{m}'$, and there exists a way of replacing ?'s in γ' and $\mathfrak{m}'$ by (possibly different) indices such that $\mathfrak{m}'$ equals $\mathfrak{m}$ and γ' is satisfied. Conversely, if $\mathfrak{C}\vdash^{s}\gamma\Rightarrow\mathfrak{m}$ and we replace all "?"'s by *the same* (existing) index, then the resulting constraint should be sufficient for the resulting message. (In particular, this is true for constraints and messages without any "?"'s.) We leave a thorough investigation of this reasoning as future work.

With patterns, we can indeed reason about var-lists using finite constraints, as the next example shows:

Example 5.7.3. We revisit Example 5.7.2, in which we considered the symbolic protocol role

$$\mathfrak{C} = \{d|_u, \{\mathsf{enc}(e|_u, f|_u)\}_{all}\}.$$

By the derivations shown in Figure 5.10, we obtain

$$\mathfrak{C}\vdash^{s} d|_u\doteq f@_{all}?|_u\Rightarrow e@_{all}?|_u; \quad \mathfrak{C}\vdash^{s} f@_{all}?|_u\Rightarrow e@_{all}?|_u.$$

Indeed, observe that all constraints from Example 5.7.2 are covered by one of these two pattern constraints. □

We can construct constraint graphs using pattern constraints; the decision procedures from Figures 5.7 and 5.8 apply with two caveats. First, as outlined above, pattern constraints are necessary and sufficient only up to patterns, so detectability and associability conclusions involving patterns should be interpreted as "maybe"s (however, undetectability and unassociability results still hold). Second, submessages of symbolic messages at different positions may be equatable. For instance, if $\mathfrak{h}(\{\mathfrak{a}\})|^{\pi} \doteq \mathfrak{h}(\{\{\mathfrak{d}\}_x, \{\mathfrak{e}\}_y\})|^{\pi}$, then both $\mathfrak{a}|^{\pi} \doteq \mathfrak{d}@_x 1|^{\pi}$ and $\mathfrak{a}|^{\pi} \doteq \mathfrak{e}@_y 1|^{\pi}$ are possible. As a consequence, the position mentioned in content equivalence nodes is no longer uniquely defined[16]. We leave a careful study of this issue as future work.

[16] In fact, the satisfiability of $\mathfrak{m} \doteq \mathfrak{m}'$ is no longer an equivalence relation.

In Chapter 7, we use var-lists to perform a symbolic analysis of privacy in Identity Mixer.

5.8 Discussion

This chapter is mainly based on our previous paper on this topic[17]. In the paper, the symbolic model is presented directly with variable-length lists, and with the multiple data subjects extension from Section 6.1. Also, the symbolic model there is described with respect to a simplified instantiated model without an information layer. In addition, there are some minor technical differences. However, the key ideas and concepts in the paper and in this chapter are the same.

[17] Veeningen et al. (2013b)

Implementing the symbolic model in an efficient way is quite subtle. In particular, when many different items or messages can be content equivalent to each other (i.e., are non-random), constraints encountered in our iterative computation (Section 5.6) can become complicated. Possibly, this latter problem can be relieved by first computing constraints that do not use content equivalence (or: that assume all content equivalence constraints to be true) and then refining the result. We leave such optimisations as future work.

In formalising variable-length lists, we have attempted to leave the original symbolic model unchanged as much as possible. In particular, we think that "pattern constraints" strike an acceptable balance between getting precise results and keeping changes to a minimum. However, some situations cannot be captured by the symbolic model (e.g., the *same* nonce occurring in each item of a var-list), and some interesting ways of deriving information are crudely approximated (cf. Example 5.7.2). Such generalisations, as well as the formal grounding of the present approach, would be an interesting direction for further study.

6
Extensions

Contents

IN THE PREVIOUS CHAPTERS, we have introduced the basic components of our framework. However, to apply the framework in practice, it may be needed to extend it; for instance, to cover additional cryptographic primitives, or additional types of personal information. In this chapter, we present several such extensions. These extensions will be used to model and analyse the case studies in Chapters 7 and 8.

Outline In this chapter:

- We propose an extension to the Personal Information Model to capture pieces of information with multiple data subjects (§6.1);
- We propose an extension to the Personal Information Model to capture predicates defined on pieces of personal information (§6.2);
- We capture the evolution of knowledge by modelling communication traces (§6.3);
- We introduce models for zero-knowledge proofs of knowledge (§6.4) and anonymous credentials (§6.5).

6.1 Multiple Data Subjects

In this section, we model pieces of information that relate to multiple data subjects. Definition 2.1.1 defined a basic PI Model, in which every piece of information has one unique data subject. However, this is not always sufficient. For instance, consider a piece of information representing a symmetric key shared between two

Description	Type	Context layer	Inf. layer data subjects
Data item	D/k	$d\vert^{\kappa}_{p_1,\ldots,p_k;\top}$	$d\vert_1,\ldots,d\vert_k$
Global identifier	I/k	$i\vert^{\kappa}_{p_1,\ldots,p_k;\varnothing}$	$i\vert_1,\ldots,i\vert_k$
Local identifier	I/k/l	$i\vert^{\kappa}_{p_1,\ldots,p_k;q_1,\ldots,q_l}$	$i\vert_1,\ldots,i\vert_k, i\vert_{;1},\ldots,i\vert_{;l}$

Table 6.1: Types of personal information in the "multiple data subjects" extension of the PI Model

parties. Using the original model, this key would either need to be considered non-personal information, or information about only one of the two parties. Similarly, a piece of information representing the final result of a chess match should be regarded as a data item about both competitors, which is not expressible in our original formalism. The alternative definition of PI Model we present in this section addresses this limitation. In addition, it allows identifiers that do not globally identify a person but only *locally* with respect to some third party: e.g., if multiple parties use sequential numbers to identify database records, then such identifiers only identify a person within the respective database. We will use pieces of information with multiple data subjects and local identifiers when symbolically analysing Identity Mixer in Section 7.8.

Modified Definition of Information Model

To allow the modelling of pieces of information with multiple data subjects, we distinguish between different *types* of personal items according to whether they are data items, global identifiers, or local identifiers; and according to the number of data subjects. For instance, type $D/2$ represents a data item with two data subjects: context data item $score\vert^{\gamma_1}_{wh,bl}$ of type $D/2$ might represent the score of a chess match γ_1 between two players $*\vert^{\gamma_1}_{wh}$ and $*\vert^{\gamma_1}_{bl}$. Context data item $score\vert^{\gamma_1}_{wh,bl}$ could map to data item $score_{1353}$, also of type $D/2$, at the information layer. The related relation $\Leftrightarrow$ can no longer be defined on information-layer items (because they have multiple data subjects): instead, it is now defined on their "data subjects", in this case $score_{1353}\vert_1$ and $score_{1353}\vert_2$.

Table 6.1 shows the different types of personal information we consider; their types; their context-layer representations; and their information-layer data subjects. Hence, an item of type D/k is a data items with k data subjects; an item of type I/k is a global identifier with k data subjects. An item of type $I/k/l$ is a local identifier with k data subjects defined with respect to l other parties. For instance, at the context layer, $id\vert^{\pi}_{u;idp}$ represents an identifier of the user represented by context $*\vert^{\pi}_{u}$, that is only guaranteed to be unique with respect to identity provider represented by context $*\vert^{\pi}_{idp}$. If this context local identifier corresponds to local identifier id_a at the information layer, then $id_a\vert_1$ and $id_a\vert_{;1}$ refer to the user and identity provider, respectively. These "data subjects" $id_a\vert_1$ and $id_a\vert_{;1}$ are used to define the related relation $\Leftrightarrow$ below.

Definition 6.1.1. A *Personal Information (PI) Model* (in the "multiple data subjects" extension) is a tuple

$$(\mathsf{O}^{\mathrm{ctx}}, \mathcal{O}^{\mathrm{inf}}, \mathsf{O}^{\mathrm{cnt}}, \Leftrightarrow, \sigma, \tau)$$

such that:

- $\mathsf{O}^{\mathrm{ctx}}$ is a set of *context personal items* of the form $v|^{\kappa}_{A;B}$. Here, v is called the *variable*; κ is called the *domain*; A (a k-length sequence of profiles) is called the *topic*; and B is called the *scope*. Each context personal information o has a type $\mathsf{t}(o)$. *Context data items* $\mathsf{D}^{\mathrm{ctx}} \subset \mathsf{O}^{\mathrm{ctx}}$ have scope $\top$ and type D/k. *Context global identifiers* $\mathsf{I}^{\mathrm{ctx}}_{g} \subset \mathsf{O}^{\mathrm{ctx}}$ have scope $\varnothing$ and type I/k. *Context local identifiers* $\mathsf{I}^{\mathrm{ctx}}_{l} \subset \mathsf{O}^{\mathrm{ctx}}$ have as scope a l-length sequence of profiles and type $I/k/l$.

- $\mathcal{O}^{\mathrm{inf}}$ is a set of *personal items* o with associated *type* $\mathsf{t}(o)$. *Data items* $\mathcal{D}^{\mathrm{inf}} \subset \mathcal{O}^{\mathrm{inf}}$ have type D/k. *Global identifiers* $\mathcal{I}^{\mathrm{inf}}_{g} \subset \mathcal{O}^{\mathrm{inf}}$ have type I/k. Local identifiers $\mathcal{I}^{\mathrm{inf}}_{l} \subset \mathcal{O}^{\mathrm{inf}}$ have type $I/k/l$.

- $\mathsf{O}^{\mathrm{cnt}} \subset \{0,1\}^*$ is a set of *contents items*;

- $\Leftrightarrow$ is an equivalence relation on data subjects $o|_i$, $o|_{;i}$ of personal items $o \in \mathcal{O}^{\mathrm{inf}}$ called the *related* relation;

- σ is a map $\mathsf{O}^{\mathrm{ctx}} \to \mathcal{O}^{\mathrm{inf}}$ such that 1) for all $o \in \mathsf{O}^{\mathrm{ctx}}$, $\mathsf{t}(\sigma(\mathsf{o})) = \mathsf{t}(\mathsf{o})$; 2) if $d_1, d_2 \in \mathsf{O}^{\mathrm{ctx}}$ from the same context have overlapping topics or scopes, then the respective data subjects of $\sigma(d_1)$ and $\sigma(d_2)$ are related[1];

- τ is a map $\mathcal{O}^{\mathrm{inf}} \to \mathsf{O}^{\mathrm{cnt}}$ such that 1) for $i, j \in \mathcal{I}^{\mathrm{inf}}_{g}$ with $\mathsf{t}(i) = \mathsf{t}(j) = I/k$: if $\tau(i) = \tau(j)$, then $i = j$; and 2) for any $i, j \in \mathcal{I}^{\mathrm{inf}}_{l}$ with $\mathsf{t}(i) = \mathsf{t}(j) = I/k/l$: if $\tau(i) = \tau(j)$ and $i|_{;m} \Leftrightarrow j|_{;m}$ for all $m \in \{1, \ldots, l\}$, then $i = j$.

[1] For instance, $\sigma(d|^{\pi}_{cl,srv;\top})|_2 \Leftrightarrow \sigma(id|^{\pi}_{srv;\varnothing})|_1$, and similarly for other combinations.

The original definition of view can be used with this modified definition of PI Model. For coalition graphs, slight (but obvious) technical changes are needed.

We now give a small example.

Example 6.1.2. Consider a PI Model modelling information about two chess players a and b and a match between them. Namely, we model a database containing an identifier and the home country of each player; and the result of a chess match between the two players.

At the context layer, the database containing the entries about the two players is modelled by context items $id|^{db}_{1;\varnothing}$, $country|^{db}_{1;\top}$, $id|^{db}_{2;\varnothing}$, and $country|^{dbn}_{2;\top}$ (where $*|^{db}_{1}$ represents player a and $*|^{db}_{2}$ represents player b). The chess match is modelled by a domain π with context items $id|^{\pi}_{white;\varnothing}$, $id|^{\pi}_{black;\varnothing}$, $score|^{\pi}_{white,black}$ (where $*|^{\pi}_{white}$ represents player a and $*|^{\pi}_{black}$ represents player b).

At the information layer, the identifiers of the two players are modelled by global identifiers $id_a, id_b \in \mathcal{I}^{\mathrm{inf}}_{g}$ of type $I/1$, and their home countries by data items $country_a, country_b \in \mathcal{D}^{\mathrm{inf}}$ of type $D/1$, respectively. The final score of the match is modelled by a data item $score_{1353} \in \mathcal{D}^{\mathrm{inf}}$ of type $D/2$ such that

$$id_a|_1 \Leftrightarrow country_a|_1 \Leftrightarrow score_{1353}|_1; \quad id_b|_1 \Leftrightarrow country_b|_1 \Leftrightarrow score_{1353}|_2.$$

A possible view on this PI Model is $V_c = (\mathsf{O}_c, \leftrightarrow_c)$, where

$$\mathsf{O}_c = \{id|^{db}_{2;\varnothing}, country|^{db}_{2;\top}, id|^{\pi}_{white;\varnothing}, id|^{\pi}_{black;\varnothing}, score|^{\pi}_{white,black}\}$$

and $*|^{db}_{2} \leftrightarrow_c *|^{\pi}_{black}$. The actor holding this view knows the database entry about player b and the information about the chess match; moreover, he knows that the player in the database and the black player of the match are the same. □

Reasoning about Multiple Data Subjects

Extending the reasoning systems of Chapters 3–5 to the multiple data subjects extension is straightforward.

When using the instantiated model from Chapter 3, we only need to change the associability rule from Definition 3.5.3 to take into account local and global identifiers. Rather than looking at the profile of an identifier, we now need to look at its topic and scope:

Definition 6.1.3. Let $*|^{\pi}_{a}$ and $*|^{\eta}_{c}$ be two contexts, and $\mathsf{i}_1, \mathsf{i}_2$ be two context identifiers. We say that $\mathsf{i}_1, \mathsf{i}_2$ *link* contexts $*|^{\pi}_{a}$ and $*|^{\eta}_{c}$ if

$$\mathsf{i}_1 = i|^{\pi}_{\{a_i\};\{b_j\}} \wedge \mathsf{i}_2 = i|^{\eta}_{\{c_i\};\{d_j\}} \wedge \forall i : *|^{\pi}_{b_i} \Leftrightarrow *|^{\eta}_{d_i} \wedge \exists j : a = a_j \wedge c = c_j.$$

Then, we define the associability relation $\leftrightarrow$ in an actor's view as the least equivalence relation on contexts such that whenever $\mathcal{C} \vdash \mathsf{i}_1 \doteq \mathsf{i}_2$ and $\mathsf{i}_1, \mathsf{i}_2$ link $*|^{\pi}_{a}$ and $*|^{\eta}_{c}$, then $*|^{\pi}_{a} \leftrightarrow *|^{\eta}_{c}$.

The example below demonstrates associability in the multiple data subjects extension:

Example 6.1.4. Consider the PI Model in the multiple data subjects extension from Example 6.1.2, in which we modelled two chess players and a match between them. Consider knowledge base

$$\mathcal{C} = \{id|^{db}_{2;\varnothing}, country|^{db}_{2;\top}, id|^{\pi}_{white;\varnothing}, id|^{\pi}_{black;\varnothing}, score|^{\pi}_{white,black}\}$$

on this PI Model, where $id|^{db}_{2;\varnothing}$ and $id|^{\pi}_{black;\varnothing}$ are two context identifiers representing the same identifier. In the view $V_c = (\mathsf{O}_c, \leftrightarrow_c)$ corresponding to $\mathcal{C}$, clearly $\mathsf{O}_c = \mathcal{C}$: all items in the knowledge base are pieces of personal information. Moreover, $*|^{db}_{2} \leftrightarrow_c *|^{\pi}_{black}$: indeed, context identifiers $id|^{db}_{2;\varnothing}$ and $id|^{\pi}_{black;\varnothing}$ link contexts $*|^{db}_{2} \leftrightarrow_c *|^{\pi}_{black}$ according to the above definition. However, context $*|^{\pi}_{white}$ is not associable to other contexts. □

The equational model (Chapter 4) and the symbolic model (Chapter 5) can be adapted similarly. For the equational model, Definition 4.3.8 changes analogously to above; for the symbolic model, the constraints graph (Section 5.5) should take into account that a single identifier can be used to link different profiles, and that a single piece of information can occur in multiple profile nodes. Currently, only our implementation of the symbolic model[2] supports this extension.

[2] Available at `http://code.google.com/p/objective-privacy/`

Discussion

In the above definitions, topics (and scopes of context local identifiers) are (ordered) sequences rather than sets. For instance, an identifier id has a well-defined "first data subject" $id|_1$ and "second data subject" $id|_2$. Context-layer representations need to respect this, e.g., if $id|^{\pi}_{a,b;\varnothing}$ instantiates id, then context $*|^{\pi}_{a}$ represents data subject $id|_1$ and context $*|^{\pi}_{b}$ represents data subject $id|_2$. For pieces of information that are essentially symmetric (such as shared keys), this model may not be accurate because an actor who sees the same information twice, may not know which topics correspond. We adopted the present definition because it is easier than one in which topics are sets. Moreover, even if information is symmetric, there is usually some asymmetry in how it is used. For instance, a key can be shared between a client and a server: in this case, we can simply adopt the convention that the first data subject is the client and the second one is the server.

In earlier work[3], this extension was presented in combination with the Symbolic Information Model.

[3] Veeningen et al. (2013b)

6.2 Attribute Predicates

We now present an extension of our PI Model formalism (cf. Definition 2.1.1) modelling boolean predicates (e.g., "this age is over 60") on pieces of information. The explicit modelling of predicates allows us to define privacy properties like "the service provider should learn that the age of the data subject is over 60, but it should not learn the actual age" at a high-level, and verify them automatically. We will use this extension in our identity management case study (Chapter 7).

Modified Definition of Information Model

We consider a fixed set of boolean *predicates* relevant to the application domain. If an identifier or data item satisfies the predicate, a *predicate item* is added to the PI Model. For instance, consider predicate $gt60$ representing that an age is over 60. If data item $d \in \mathcal{D}^{\text{inf}}$ is an age over 60, we add predicate item $d?_{gt60}$ to the PI Model. If d has context-layer representation $d|^{\pi}_{k}$, then $d?_{gt60}$ has context-layer representation $d?_{gt60}|^{\pi}_{k}$. If predicate item $d?_{gt60}$ does not exist, then either d it is not an age, or it is an age below 60. Formally:

Definition 6.2.1. A *Personal Information (PI) Model* (in the "attribute predicates" extension) is a tuple

$$(\mathsf{O}^{\text{ctx}}, \mathcal{O}^{\text{inf}}, \mathsf{O}^{\text{cnt}}, \Leftrightarrow, \sigma, \tau)$$

such that:

- O^{ctx} is a set of *context personal items* of the form $v|^{\kappa}_{a}$. Here, v is called the *variable*, κ is called the *domain*, and a is called the *profile*. O^{ctx} is partitioned into *context data items* $\mathsf{D}^{\text{ctx}} \subset \mathsf{O}^{\text{ctx}}$, *context*

identifiers $\mathsf{I}^{\text{ctx}} \subset \mathsf{O}^{\text{ctx}}$, and *context predicate item*

$$\mathsf{R}^{\text{ctx}} = \{d?_p|_k^\pi \mid d|_k^\pi \in \mathsf{D}^{\text{ctx}} \cup \mathsf{I}^{\text{ctx}}, \sigma(d|_k^\pi)?_p \in \mathcal{R}^{\text{inf}}\} \subset \mathsf{O}^{\text{ctx}}$$

(with $\mathcal{R}^{\text{inf}}$ defined below);

- $\mathcal{O}^{\text{inf}}$ is a set of *personal items*, partitioned into sets $\mathcal{D}^{\text{inf}} \subset \mathcal{O}^{\text{inf}}$ of *data items*, $\mathcal{I}^{\text{inf}} \subset \mathcal{O}^{\text{inf}}$ of *identifiers*, and $\mathcal{R}^{\text{inf}} \subset \mathcal{O}^{\text{inf}}$ of *predicate items* of the form $p?_{pr}$ with $p \in \mathcal{D}^{\text{inf}} \cup \mathcal{I}^{\text{inf}}$ and pr a *predicate*;
- $\mathbb{O}^{\text{cnt}} \subset \{0,1\}^*$ is a set of *contents items*;
- $\Leftrightarrow$ is an equivalence relation on $\mathcal{O}^{\text{inf}}$ called the *related* relation, such that $p?_{pr} \Leftrightarrow p$ for all p, pr;
- σ is a map $\mathsf{O}^{\text{ctx}} \to \mathcal{O}^{\text{inf}}$ such that 1) $\sigma(\mathsf{I}^{\text{ctx}}) \subset \mathcal{I}^{\text{inf}}$, $\sigma(\mathsf{D}^{\text{ctx}}) \subset \mathcal{D}^{\text{inf}}$, and $\sigma(d?_{pr}|_k^\pi) = \sigma(d|_k^\pi)?_{pr}$; 2) $\sigma(x|_k^\kappa) \Leftrightarrow \sigma(y|_k^\kappa)$ for all $x|_k^\kappa, y|_k^\kappa$;
- τ is a map $\mathcal{O}^{\text{inf}} \to \mathbb{O}^{\text{cnt}}$ such that 1) for any identifiers $i, j \in \mathcal{I}^{\text{inf}}$: if $\tau(i) = \tau(j)$, then $i = j$; 2) for any predicate items $p?_{pr}, q?_{pr}$, $\tau(p?_{pr}) = \tau(q?_{pr})$;

Compared to the original definition of PI Model (Definition 2.1.1), we additionally define predicate items at the context and information layers, and define how $\Leftrightarrow$, σ and τ operate on them. Namely, predicate items are related to the personal items they are a predicate about (fourth bullet point); predicate items at the context and information layer are consistent (fifth bullet point); and the contents of predicate items are independent from which item they are a predicate about (sixth bullet point).

The original definitions of views and coalition graphs can be used with this modified definition of PI Model.

We now give a small example.

Example 6.2.2. Consider a PI Model consisting of three data items $\mathcal{D}^{\text{inf}} = \{age_c, age_d, aptno_c\}$. Data items age_c, age_d represent the ages of two data subjects; $aptno_c$ represents the number of the apartment that the first data subject lives in. Consider predicate $gt60$ representing that an age is over 60. If age_c, age_d satisfy the predicate $gt60$, then $\mathcal{R}^{\text{inf}} = \{age_c?_{gt60}, age_d?_{gt60}\}$. By definition, $\tau(age_c?_{gt60}) = \tau(age_d?_{gt60})$. Although age_c and age_d both satisfy predicate $gt60$, they may not have the same contents, e.g., $\tau(age_c) = \text{“60”} \neq \text{“62”} = \tau(age_d)$. Similarly, although age_c and $aptno_c$ do not both satisfy predicate $gt60$, they may still have the same contents (because information $aptno_c$ is of a different kind).

Suppose that $age_c, age_d, aptno_c$ have context-layer representations $\mathsf{D}^{\text{ctx}} = \{age|_u^\pi, age|_u^\eta, aptno|_u^\pi\}$, respectively. Then the set of context predicate items is $\mathsf{R}^{\text{ctx}} = \{age?_{gt60}|_u^\pi, age?_{gt60}|_u^\eta\}$. □

Reasoning about Attribute Predicates

To extend the reasoning system from Chapter 3 to the attribute predicates extension, we need to modify the definitions of derivability

$$\dfrac{\dfrac{\dfrac{}{\mathcal{C} \vdash \mathsf{h}(age|_u^\pi)}(\vdash\mathbf{0}) \quad \dfrac{}{\mathcal{C} \vdash aptno|_u^\pi}(\vdash\mathbf{0})}{\mathcal{C} \vdash age|_u^\pi}(\vdash\mathbf{E})}{\mathcal{C} \vdash age?_{gt60}|_u^\pi}(\vdash\mathbf{R})$$

Figure 6.1: Derivation of predicate items (Example 6.2.3)

(Definition 3.3.4), and direct equatability (Definition 3.5.2). For derivability, we add an inference rule

$$\dfrac{\mathcal{C} \vdash d|_k^\pi}{\mathcal{C} \vdash d?_p|_k^\pi}\,(d?_p|_k^\pi \in \mathsf{R}^{\mathrm{ctx}})\ \ (\vdash\mathbf{R})$$

to the deductive system of Figure 3.1, allowing an actor to derive predicates about attributes he knows. If an actor can equate two attributes, we also allow him to equate their predicates; this knowledge may be used to detect predicates in new contexts. Formally, we change Definition 3.5.2 so that, if $\mathcal{C} \vdash d|_k^\pi \doteq e|_l^\eta$ and $d?_p|_k^\pi, e?_p|_l^\eta \in \mathsf{R}^{\mathrm{ctx}}$, then also $\mathcal{C} \vdash d?_p|_k^\pi \doteq e?_p|_l^\eta$.

We demonstrate reasoning about attribute predicates with an example.

Example 6.2.3. Consider again the PI Model in the attribute predicates extension from Example 6.2.2. This PI Model consists of three data items $\mathcal{D}^{\mathrm{inf}} = \{age_c, age_d, aptno_c\}$ with $\tau(age_c) = \tau(aptno_c) \neq \tau(age_d)$ and predicates $\mathcal{R}^{\mathrm{inf}} = \{age_c?_{gt60}, age_d?_{gt60}\}$; and respective context-layer representations

$$\mathsf{D}^{\mathrm{ctx}} = \{age|_u^\pi, age|_u^\eta, aptno|_u^\pi\};\quad \mathsf{R}^{\mathrm{ctx}} = \{age?_{gt60}|_u^\pi, age?_{gt60}|_u^\eta\}.$$

Now consider knowledge base $\mathcal{C} = \{\mathsf{h}(age|_u^\pi), aptno|_u^\pi, age?_{gt60}|_u^\eta\}$. We have that $\mathcal{C} \vdash age?_{gt60}|_u^\pi$ by the derivation shown in Figure 6.1. Intuitively, the actor can derive $age|_u^\pi$ by reconstruction, and then find out that it is greater than 60. In fact, the view $V = (\mathsf{O}, \leftrightarrow)$ corresponding to $\mathcal{C}$ consists of $\mathsf{O} = \{age|_u^\pi, aptno|_u^\pi, age?_{gt60}|_u^\pi, age?_{gt60}|_u^\eta\}$ and no associable contexts. □

Discussion

Above, we have considered the present extension only for the rule-based framework of Chapter 3. Our tool[4] for the formal analysis of privacy in communication protocols implements this extension.

[4] Available at `http://code.google.com/p/objective-privacy/`

However, we do not consider the extension for the equational framework of Chapter 4 or the symbolic framework of Chapter 5. It should be possible to define attribute predicates in the equational framework of Chapter 4, but this would make it harder to automatically compute views. Without attribute predicates, detectability in our model is equivalent to deducibility plus a list of non-static equivalences. Because attribute predicates introduce dependencies

between equatability of different pairs of items, the correspondence between detectability and equational properties would become more complicated. Similarly, defining attribute predicates in the symbolic framework of Chapter 5 may be possible, but would complicate the decision procedure for detectability (Figure 5.7). We leave the definition in the equational and symbolic models as future work.

In earlier work[5], this extension was presented in combination with the Personal Information Model.

[5] Veeningen et al. (2014)

6.3 States, Traces, and System Evolution

So far, we have shown how the knowledge about personal information of actors and coalitions follows from sets of known messages. However, instead of specifying these sets, it is more intuitive to model a scenario by specifying the initial knowledge of actors and the messages they exchange. Moreover, by modelling the exchange of messages, we can verify that modelled initial knowledge of the actors is sufficient to send each message, which ensures that we get a complete picture of actor knowledge. For instance, if a protocol involves an actor sending an encryption randomised with a fresh nonce, we can check that the actor knew the nonce before sending the message – which is relevant, e.g., for checking that he can derive the plaintext using a reconstruction rule. In this section, we introduce a light-weight formalism for the exchange of messages by actors, and show how it is used to ensure that all relevant knowledge is modelled.

6

Our model of system evolution consists of *states*, which capture the knowledge of all actors in a system at a certain point in time; and *traces*, which model communication steps that result in new states. Each actor has his own knowledge base. The knowledge about personal information by an actor, captured by his view, follows from his knowledge base. The knowledge of coalitions of actors follows from the union of their respective knowledge bases:

Definition 6.3.1. Let $\mathcal{A}$ be a set of actors, and I an Information Model.

- A *state* is a collection $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ of knowledge bases about I;
- The *view of actor* $a \in \mathcal{A}$ *in state* $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ is the view corresponding to knowledge base $\mathcal{C}_a$ (Definition 3.5.3);
- The *view of coalition* $\{a_1, \ldots, a_k\} \subset \mathcal{A}$ *of actors in state* $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ is the view corresponding to knowledge base $\mathcal{C}_{a_1} \cup \ldots \cup \mathcal{C}_{a_k}$.

We assume that Information Model I is fixed. That is, changes in knowledge during the system evolution are modelled by different states of the same Information Model I.

A *trace* is a sequence of communication steps, modelled by *transmissions*. A *transmission* $a(\mathsf{id}_a) \to b(\mathsf{id}_b) : \mathsf{m}$ models that actor a uses communication address id_a (typically, an identifier corresponding to

his protocol role) to send message m to actor b with communication address id_b. A *trace* $\mathfrak{T}$ is a sequence $\mathsf{t}_1;\cdots;\mathsf{t}_k$ of transmissions. States *evolve* by traces so that the actors involved learn the messages they exchange:

Definition 6.3.2. An *evolution* from state $\{\mathcal{C}_x^0\}_{x\in\mathcal{A}}$ into state $\{\mathcal{C}_x^k\}_{x\in\mathcal{A}}$ by trace $\mathsf{t}_1;\cdots;\mathsf{t}_k$ is a series of steps (let $\mathsf{t}_i = a_i(\mathsf{id}_i) \to b_i(\mathsf{id}_i') \ : \ \mathsf{m}_i$):

$$\{\mathcal{C}_x^0\}_{x\in\mathcal{A}} \xrightarrow{\mathsf{t}_1} \{\mathcal{C}_x^1\}_{x\in\mathcal{A}} \xrightarrow{\mathsf{t}_2} \cdots \xrightarrow{\mathsf{t}_n} \{\mathcal{C}_x^n\}_{x\in\mathcal{A}},$$

where $\mathcal{C}_a^i = \mathcal{C}_a^{i-1} \cup \{\mathsf{id}_i, \mathsf{id}_i', \mathsf{m}_i\}$; $\mathcal{C}_b^i = \mathcal{C}_b^{i-1} \cup \{\mathsf{id}_i, \mathsf{id}_i', \mathsf{m}_i\}$; and $\mathcal{C}_z^i = \mathcal{C}_z^{i-1}$ for all other actors $z \in \mathcal{A} \setminus \{a, b\}$.

Hence, from a transmission $a(\mathsf{id}_a) \to b(\mathsf{id}_b) \ : \ \mathsf{m}$, actors a and b learn messages m as well as the communication addresses id_a, id_b; other actors learn nothing.

The following example demonstrates traces, states, and transmissions.

Example 6.3.3. We revisit Example 3.5.1 (page 44), in which a client *cli* and server *srv* exchange personal information about a data subject. We model a complete system evolution as a trace executed from an initial state.

We are interested in the knowledge of two actors $\mathcal{A} = \{cli, srv\}$: the client and server. The initial state $\{\mathcal{C}_x^0\}_{x\in\mathcal{A}}$ captures their initial knowledge. This should include all information needed for the communication to take place; so in particular, the client needs to know the communication address of the server, and the server needs to know his private key. Analogously to Example 3.5.1, for the client we take:

$$\mathcal{C}_{cli}^0 = \{nm|_{12}^{ab}, teln|_{12}^{ab}, nm|_4^{ab}, id|_4^{ab}, skey|_{\cdot}^{\cdot}, \mathsf{pk}(k^-|_{srv}^{\cdot}), ip|_{me}^{\cdot}, ip|_{srv}^{\cdot}\},$$

where $ip|_{me}^{\cdot}$, $ip|_{srv}^{\cdot}$ represent the communication addresses of the client and server, respectively. Similarly, for the server, we take:

$$\mathcal{C}_{srv}^0 = \{col1|_1^{db}, key|_1^{db}, col1|_2^{db}, key|_2^{db}, n|_{\cdot}^{\cdot}, skey|_{\cdot}^{\cdot}, k^-|_{srv}^{\cdot}, ip|_{srv}^{\cdot}\},$$

where $n|_{\cdot}^{\cdot}$ is the nonce from the server's reply. Communication consists of a query by the client, and a response by the server (See Example 3.5.1), formalised by the following trace:

$$cli(ip|_{cli}^{\pi}) \to srv(ip|_{srv}^{\pi}) \ : \mathsf{enc}(id|_{su}^{\pi}, shkey|_{\cdot}^{\pi});$$
$$srv(ip|_{srv}^{\pi}) \to cli(ip|_{cli}^{\pi}) \ : \mathsf{enc}(\{id|_{su}^{\pi}, \mathsf{sig}(\{age|_{su}^{\pi}, n|_{\cdot}^{\pi}\}, k^-|_{srv}^{\pi})\}, shkey|_{\cdot}^{\pi}).$$

State $\{\mathcal{C}_x^0\}_{x\in\mathcal{A}}$ evolves by this trace into state $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ shown in Figure 6.2. Note that, apart from the communication addresses, $\mathcal{C}_{cli}$ is as in Example 3.5.1. Hence, the knowledge of *cli*, *srv*, and coalition $\{cli, srv\}$ about Alice and Bob in this state is as in Example 3.5.4. □

We now show how to verify that all knowledge required for the communication in a trace has been modelled. For this, we need to model whether a context item has occurred in communication before. When an actor a initiates a protocol instance π in state $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$,

Figure 6.2: State after system evolution in Example 6.3.3

$$\mathcal{C}_{cli} = \{nm|^{ab}_{12}, teln|^{ab}_{12}, nm|^{ab}_{4}, id|^{ab}_{4}, skey|^{\cdot}_{\cdot}, \mathsf{pk}(k^{-}|^{\cdot}_{srv}), ip|^{\cdot}_{me}, ip|^{\cdot}_{srv}, ip|^{\pi}_{cli}, ip|^{\pi}_{srv}, \mathsf{enc}(id|^{\pi}_{su}, shkey|^{\pi}_{\cdot}), \mathsf{enc}(\{id|^{\pi}_{su}, \mathsf{sig}(\{age|^{\pi}_{su}, n|^{\pi}_{\cdot}\}, k^{-}|^{\pi}_{srv})\}, shkey|^{\pi}_{\cdot})\}$$

$$\mathcal{C}_{srv} = \{col1|^{db}_{1}, key|^{db}_{1}, col1|^{db}_{2}, key|^{db}_{2}, n|^{\cdot}_{\cdot}, skey|^{\cdot}_{\cdot}, k^{-}|^{\cdot}_{srv}, ip|^{\cdot}_{srv}, ip|^{\pi}_{cli}, ip|^{\pi}_{srv}, \mathsf{enc}(id|^{\pi}_{su}, shkey|^{\pi}_{\cdot}), \mathsf{enc}(\{id|^{\pi}_{su}, \mathsf{sig}(\{age|^{\pi}_{su}, n|^{\pi}_{\cdot}\}, k^{-}|^{\pi}_{srv})\}, shkey|^{\pi}_{\cdot})\}$$

no communication in the protocol instance has taken place yet, so the state does not contain context items with domain π. Hence, to check whether a can send message $\mathsf{m}|^{\pi}$, we cannot just verify if $\mathcal{C}_a \vdash \mathsf{m}|^{\pi}$. Instead, we need to model that the actor "instantiates" the context items in $\mathsf{m}|^{\pi}$ by items from other domains. On the other hand, if actor b wants to reply to message $\mathsf{m}|^{\pi}$, then he no longer has the freedom to instantiate the contents of context items from $\mathsf{m}|^{\pi}$. In the former case, we call the context items *undetermined*; in the latter case, we call them *determined*:

Definition 6.3.4. Let $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ be a state. We say that $\mathsf{p} \in \mathsf{P}^c$ is *determined* in $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ if, for some $a \in \mathcal{A}$ and $\mathsf{m} \in \mathcal{C}_a$, p occurs in m.[6] Otherwise, we say that p is *undetermined*.

[6] Or, when using the attribute predicates extension from the previous section, if p is a predicate $d?_{pr}|^{\pi}_{k}$ of a determined context item $d|^{\pi}_{k}$

We now formalise when an actor has sufficient knowledge in a certain state to send a certain message $\mathsf{m}|^{\pi}$. The actor can instantiate any undetermined items in $\mathsf{m}|^{\pi}$, but needs to respect the existing instantiation of determined items in $\mathsf{m}|^{\pi}$. Intuitively, we formalise "sufficient knowledge" by requiring that message $\mathsf{m}|^{\pi}$ does not change the view of the actor in any essential way. Namely, we require that the actor can derive a message n containing the same information as m such that everything he can learn from $\mathsf{m}|^{\pi}$, he can also learn from n. For instance, if $\mathsf{m}|^{\pi}$ contains a piece of information that the actor can associate to some context, then the actor should also be able to associate the corresponding piece of information in n to that same context. Because he can derive n, he does not learn anything from $\mathsf{m}|^{\pi}$. Formally:

Definition 6.3.5. Let $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ be a state, and $a \in \mathcal{A}$ an actor. Context message m is *determinable* by a in $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ if there exists a context message n such that $\mathcal{C}_a \vdash \mathsf{n}$, and the following conditions hold:

1. For all z, $\sigma(\mathsf{n}@z) = \sigma(\mathsf{m}@z)$;
2. If $\mathsf{m}@z$ is determined, then $\mathcal{C} \vdash \mathsf{m}@z \doteq \mathsf{n}@z$;
3. If $\mathsf{m}@z_1 = \mathsf{m}@z_2$, then $\mathcal{C} \vdash \mathsf{n}@z_1 \doteq \mathsf{n}@z_2$;
4. If $\mathsf{m}@z = d|^{\kappa}_{k}$ $(k \neq \cdot)$, $\mathsf{n}@z = d'|^{\pi}_{l}$ $(l \neq \cdot)$, and some $i|^{\kappa}_{k} \in \mathsf{I}^{\mathrm{ctx}}$ is determined, then $*|^{\kappa}_{k} \leftrightarrow_a *|^{\pi}_{l}$;
5. If $\mathsf{m}@z_1 = d|^{\kappa}_{k}$, $\mathsf{m}@z_2 = d'|^{\kappa}_{k}$ $(k \neq \cdot)$, $\mathsf{n}@z_1 = e|^{\pi}_{l}$ $(l \neq \cdot)$, and $\mathsf{n}@z_2 = f|^{\eta}_{m}$ $(m \neq \cdot)$, then $*|^{\pi}_{l} \leftrightarrow_a *|^{\eta}_{m}$.

Condition 1 states that message n should contain the same information as m. Condition 2 states the actor can only replace determined items by other items that he can equate them to; condition 3 states that he should replace items consistently. Conditions 4 and 5 ensure that the actor cannot learn new associations by using n as m: condition 4 ensures that he does not learn associations between new and existing information; condition 5 ensures that he does not learn associations between new information.

The following example demonstrates determinability:

Example 6.3.6. Consider the state $\{\mathcal{C}_x^0\}_{x\in\mathcal{A}}$ from Example 6.3.3. The client's message $\mathsf{m} = \mathsf{enc}(id|_{su}^{\pi}, shkey|_{\cdot}^{\pi})$ is determinable by cli in this state. Namely, take $\mathsf{n} = \mathsf{enc}(id|_4^{ab}, skey|_{\cdot})$: this message trivially satisfies conditions 1–5 of the definition.

Also, the server's reply to this message is determinable. Namely, consider the state $\{\mathcal{C}_x^1\}_{x\in\mathcal{A}}$ that $\{\mathcal{C}_x^0\}_{x\in\mathcal{A}}$ evolves into. The server's knowledge base in $\{\mathcal{C}_x^1\}_{x\in\mathcal{A}}$ is:

$$\mathcal{C}_{srv}^1 = \{col1|_1^{db}, key|_1^{db}, col1|_2^{db}, key|_2^{db}, n|_{\cdot}, skey|_{\cdot}, k^-|_{srv}^{\cdot}, ip|_{srv}^{\cdot}, \\ ip|_{cli}^{\pi}, ip|_{srv}^{\pi}, \mathsf{enc}(id|_{su}^{\pi}, shkey|_{\cdot}^{\pi})\},$$

and the server's reply is:

$$\mathsf{m} = \mathsf{enc}(\{id|_{su}^{\pi}, \mathsf{sig}(\{age|_{su}^{\pi}, n|_{\cdot}^{\pi}\}, k^-|_{srv}^{\pi})\}, shkey|_{\cdot}^{\pi}).$$

Indeed, one can verify that

$$\mathsf{n} = \mathsf{enc}(\{id|_{su}^{\pi}, \mathsf{sig}(\{col1|_1^{db}, n|_{\cdot}\}, k^-|_{srv}^{\cdot})\}, shkey|_{\cdot}^{\pi})$$

satisfies the conditions from the above definition. Namely, n contains the same information as m (condition 1); all determined items from m are the same in n (condition 2); all undetermined items have (trivially) been replaced consistently (condition 3); and $*|_1^{db} \leftrightarrow_{srv} *|_{su}^{\pi}$, i.e., the message contains only known associations (condition 4). Condition 5 holds trivially because no two different context items from the same context are instantiated. □

The validity of a trace is defined by verifying determinability at every communication step. Namely, in order to send a message, an actor should be able to determine the message as well as the communication identifiers used:

Definition 6.3.7. Let $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ be a state, and $\mathsf{t} = a(\mathsf{id}_a) \to b(\mathsf{id}_b) : \mathsf{m}$ a transmission. We say that t is *valid* in $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ if the message $\{\mathsf{id}_a, \mathsf{id}_a, \mathsf{m}\}$ is determinable by a in $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$. Trace $\mathsf{t}_1; \cdots; \mathsf{t}_k$ is *valid* in state $\{\mathcal{C}_x^0\}_{x\in\mathcal{A}}$ if, in the evolution

$$\{\mathcal{C}_x^0\}_{x\in\mathcal{A}} \xrightarrow{\mathsf{t}_1} \{\mathcal{C}_x^1\}_{x\in\mathcal{A}} \xrightarrow{\mathsf{t}_2} \cdots \xrightarrow{\mathsf{t}_n} \{\mathcal{C}_x^n\}_{x\in\mathcal{A}},$$

each transmission t_i is valid in respective state $\{\mathcal{C}_x^{i-1}\}_{x\in\mathcal{A}}$.

The following example shows validity of transmissions and traces.

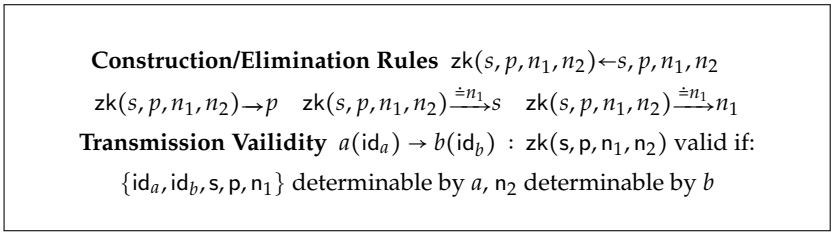


Figure 6.3: Formal model of zero-knowledge proofs; reconstruction rule is implicit

Example 6.3.8. Consider the trace given in Example 6.3.3. In Example 6.3.6, we showed determinability of the two messages transmitted in the trace; this argument can be easily extended to include determinability of the communication identifiers. Therefore, the two transmissions are valid in their respective states. Hence, the given trace is valid. □

Our tool[7] for the formal analysis of privacy in communication protocols implements this extension. This extension appeared in previous work[8].

[7] Available at `http://code.google.com/p/objective-privacy/`

[8] Veeningen et al. (2014)

6.4 Zero-Knowledge Proofs of Knowledge

In this section, we propose a model of zero-knowledge proofs of knowledge for the rule-based framework of Chapters 3 and 5. A zero-knowledge proof of knowledge is a cryptographic protocol between two parties: the *prover* and the *verifier*. At the start of the protocol, the prover and verifier both know some public information (e.g., a public key pk). In the protocol, the prover convinces the verifier that he knows some secret information which satisfies a certain property with respect to the public information (e.g., the secret key corresponding to pk), without revealing any additional information about this secret. See Quisquater et al.[9] for a *very* high-level introduction.

[9] Quisquater et al. (1989)

We model a particular family of zero-knowledge proofs called Σ-protocols[10]. A Σ-protocol consists of three messages: first, a commitment to the secret information by the prover; second, a random challenge by the verifier; and third, a response to this challenge by the prover. We model the protocol transcript of a Σ-protocol, i.e., the three messages combined, with function symbol $\mathsf{zk}/4$. Namely, $\mathsf{zk}(\mathsf{s},\mathsf{p},\mathsf{n}_1,\mathsf{n}_2)$ represents the transcript of a Σ-protocol with secret s, public information p, randomness n_1 for the commitment of the prover, and challenge n_2 of the verifier. For instance, the above zero-knowledge proof of knowledge of the private key corresponding to a public key can be modelled as $\mathsf{zk}(k^-|^{\pi}_{u}, \mathsf{pk}(k^-|^{\pi}_{u}), n_1|^{\pi}_{.}, n_2|^{\pi}_{.})$.

[10] Cramer (1997)

The construction and elimination rules for zk are shown in Figure 6.3. Namely, a zero-knowledge proof of knowledge can be constructed from its parts. The public information can be derived from the proof, that is, anybody can see what public information is used in the proof. The private information can be derived using the prover's nonce and, by the visible failure assumption, also the prover's nonce can be tested. We argue below that these rules are

accurate assuming that the randomness from the prover and verifier is not re-used. That is, the randomness can (and should) occur as context items in their respective knowledge bases, but it should not occur in other messages (in particular, in other zero-knowledge proofs).

Note that the zk symbol does *not* capture how the secret relates to the public information. Instead, the relation between the secret and public information is modelled by their context-layer representations. For instance, $\mathsf{zk}(k^-|_u^\pi, \mathsf{pk}(k^-|_u^\pi), n_1|_\cdot^\pi, n_2|_\cdot^\pi)$ models a proof that the prover knows the private key corresponding to the given public key because both are modelled by the same context item $k^-|_u^\pi$.

The following example shows knowledge bases containing zero-knowledge proofs of knowledge.

Example 6.4.1. Consider actor a proving to actor b that he knows the items $pw|_u^\pi, n|_\cdot^\pi$ in a cryptographic hash $\mathsf{h}(\{pw|_u^\pi, n|_\cdot^\pi\})$. Let $n_a|_\cdot^\pi$, $n_b|_\cdot^\pi$ denote the randomness of a and b in the proof, respectively. Then:

$$\{n_a|_\cdot^\pi, \mathsf{zk}(\{pw|_u^\pi, n|_\cdot^\pi\}, \mathsf{h}(\{pw|_u^\pi, n|_\cdot^\pi\}), n_a|_\cdot^\pi, n_b|_\cdot^\pi)\} \subset \mathcal{C}_a$$
$$\{n_b|_\cdot^\pi, \mathsf{zk}(\{pw|_u^\pi, n|_\cdot^\pi\}, \mathsf{h}(\{pw|_u^\pi, n|_\cdot^\pi\}), n_a|_\cdot^\pi, n_b|_\cdot^\pi)\} \subset \mathcal{C}_b.$$

That is, both parties know the transcript of the proof and their own randomness. From this transcript, a can derive the secret $\{pw|_u^\pi, n|_\cdot^\pi\}$ using his randomness; b can only derive the public information $\mathsf{h}(\{pw|_u^\pi, n|_\cdot^\pi\})$. □

We model the evolution of a system due to a zero-knowledge proof of knowledge by adapting the formalism of Section 6.3. In particular, transmission $a(\mathsf{id}_a) \to b(\mathsf{id}_b) : \mathsf{zk}(\mathsf{s}, \mathsf{p}, \mathsf{n}_1, \mathsf{n}_2)$ denotes a zero-knowledge proof of knowledge in which a is the prover and b is the verifier[11]. For this transmission to be valid (cf. Definition 6.3.7), $\{\mathsf{id}_a, \mathsf{id}_b, \mathsf{s}, \mathsf{p}, \mathsf{n}_1\}$ should be determinable by a, and n_2 should be determinable by b; see Figure 6.3.

[11] Note that, according to the definition of Section 6.3, this transmission would denote a simulating the protocol itself and sending the transcript to b.

We now show a trace with a zero-knowledge proof of knowledge.

Example 6.4.2. Let us continue Example 6.4.1. Consider a scenario in which a first sends hash $\mathsf{h}(\{pw|_u^\pi, n|_\cdot^\pi\})$ to b; and then proves that he knows the corresponding plaintext.

In this scenario, a initially needs to know the plaintext and randomness for the proof, and the identifiers used for communication. b just needs to know his randomness for the proof:

$$\mathcal{C}_a^0 = \{pw|_a^\cdot, n|_\cdot^\cdot, n_a|_\cdot^\cdot, ip|_a^\cdot, ip|_b^\cdot\}; \quad \mathcal{C}_b^0 = \{n_b|_\cdot^\cdot, ip|_b^\cdot\}$$

Given the state $\{\mathcal{C}_x^0\}_{x\in\{a,b\}}$ consisting of these two knowledge bases, the following (valid) trace models the scenario:

$$a(ip|_u^\pi) \to b(ip|_{srv}^\pi) : \mathsf{h}(\{pw|_u^\pi, n|_\cdot^\pi\});$$
$$a(ip|_u^\pi) \to b(ip|_{srv}^\pi) : \mathsf{zk}(\{pw|_u^\pi, n|_\cdot^\pi\}, \mathsf{h}(\{pw|_u^\pi, n|_\cdot^\pi\}), n_a|_\cdot^\pi, n_b|_\cdot^\pi).$$

Indeed, in the resulting state $\{\mathcal{C}_x^2\}_{x\in\{a,b\}}$, a can derive $pw|_u^\pi$ but b can just derive $\mathsf{h}(\{pw|_u^\pi, n|_\cdot^\pi\})$. □

We now argue why the above model is a reasonable formalisation of Σ-protocols by considering the classical zero-knowledge proof of knowledge of a discrete logarithm from Schnorr[12]. Consider a group $\mathcal{G}$ with generator g in which the *discrete logarithm assumption* is satisfied. That is, given g and random group element g^x, it is infeasible to determine x. (This property can be exploited to build a public key cryptosystem in which values of g^x are public keys, and the corresponding values of x are private keys.) Using the Schnorr proof (Figure 6.4), a prover can convince a verifier who just knows g^x, that he knows the corresponding value x, without revealing x itself. The prover first computes a random u and sends a commitment g^u to u to the verifier. The verifier responds with a random challenge c. The prover calculates response $r = u + cx$. The verifier convinces himself that the prover indeed knows the secret x by checking that $g^r = ah^c$ using the response, commitment and public information. The prover can only calculate a valid response if he knows the secret; also, the response does not reveal any information about x.[13]

[12] Schnorr (1989)

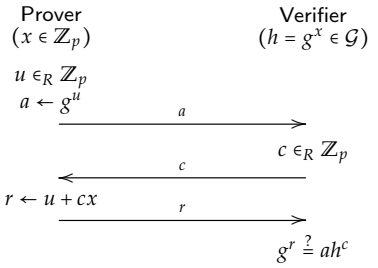


Figure 6.4: Schnorr proof of knowledge of discrete logarithm in group $\mathcal{G}$ with generator g, $|\mathcal{G}| = p$ (Source: Schoenmakers (2014))

[13] Schnorr (1989)

In a setting where public/private key pairs are pairs (g^x, x) in the above group $\mathcal{G}$, the above Schnorr proof may be modelled as $\mathsf{zk}(k^-|^\pi_{cli}, \mathsf{pk}(k^-|^\pi_{cli}), n_1|^\pi_., n_2|^\pi_.)$, where $k^-|^\pi_{cli}$, $n_1|^\pi_.$, and $n_2|^\pi_.$ are pieces of information with contents x, u, and c, respectively, and $\mathsf{pk}(k^-|^\pi_{cli})$ is a context message with contents $h = g^x$. The contents of the zero-knowledge proof are a concatenation of the values a, c, and r from the protocol transcript. Let us consider what knowledge can be obtained from this protocol transcript. The public information h can be computed from the protocol transcript as $h = (ag^{-r})^{1/c}$,[14] so:

$$\mathsf{zk}(k^-|^\pi_{cli}, \mathsf{pk}(k^-|^\pi_{cli}), n_1|^\pi_., n_2|^\pi_.) \rightarrow \mathsf{pk}(k^-|^\pi_{cli}).$$

[14] Provided $c \neq 0$, which is true with high probability

Moreover, the private information can be computed from the transcript given the prover's randomness u: $x = (r - u)c^{-1}$, i.e.,

$$\mathsf{zk}(k^-|^\pi_{cli}, \mathsf{pk}(k^-|^\pi_{cli}), n_1|^\pi_., n_2|^\pi_.) \xrightarrow{\doteq n_1|^\pi_.} k^-|^\pi_{cli}.$$

The prover's randomness can be tested (by recomputing g^u) and the verifier's randomness can be derived directly (because it occurs in the transcript). Because we assume that this randomness does not occur as submessage of any other message of actors, we do not need to include a rule to derive the verifier's randomness; the rule to derive the prover's randomness is required by the visible failure assumption of our model of cryptographic primitives.

Generalising the above reasoning, we obtain the model shown in Figure 6.3. Note that, under the above assumptions, this model represents the "worst case" of what information actors can derive from the transcript. Namely, nobody except the prover in a zero-knowledge proof of knowledge should learn anything about the secret. Hence, there should be no rule to derive the secret that does not require the prover's nonce.[15] For the same reason, there can be no rule to derive the prover's nonce without knowing its contents. On the other hand, a zero-knowledge proof of knowledge does not

[15] In particular, there is no rule to test the secret from the proof. In the Schnorr example, a guess for the secret can still be verified by constructing the public value from it and testing that. However, if multiple secrets give the same public value, then nobody except the prover should know which one of them was used.

protect the public information; and indeed, we have a rule to derive it from the transcript without any auxiliary messages.

For validity of the transmission $a(\mathsf{id}_a) \to b(\mathsf{id}_b) : \mathsf{zk}(\mathsf{s}, \mathsf{p}, \mathsf{n}_1, \mathsf{n}_2)$, it is clear that the prover needs to know $\{\mathsf{id}_a, \mathsf{id}_b, \mathsf{s}, \mathsf{n}_1\}$ and the verifier needs to know n_2. In the particular case of the Schnorr proof, the prover does not use the public information p; however, the prover still knows the public information because it follows from the private information. On the other hand, in cases where the public information does not follow from the private information, the prover *does* use the public information in the protocol. Hence, for validity, we also demand that the prover can determine p.

For example zero-knowledge proofs, see the identity management case study in Chapter 7.

Discussion

As mentioned above, we model the relation between the secret and public information not as part of the zk primitive, but by the choice of context-layer representations. This means we model actors who know what properties are proven in a zero-knowledge proof, i.e., we do not consider attackers who intercept a zero-knowledge proof and try to gain knowledge by investigating which property is proven. It also means that, in our model, zero-knowledge proofs with the same structure of public and secret information but with a different property may be content equivalent. For instance, if $k^-|_a^\pi \doteq k^-|_b^\pi$, then the following two context messages are content equivalent:

$$\mathsf{zk}(\{k^-|_a^\pi, k^-|_b^\pi\}, \{\mathsf{pk}(k^-|_a^\pi), \mathsf{pk}(k^-|_b^\pi)\}, n_1|_\cdot^\pi, n_2|_\cdot^\pi)\})$$
$$\mathsf{zk}(\{k^-|_b^\pi, k^-|_a^\pi\}, \{\mathsf{pk}(k^-|_a^\pi), \mathsf{pk}(k^-|_b^\pi)\}, n_1|_\cdot^\pi, n_2|_\cdot^\pi)\}).$$

In practice, however, the protocol transcripts they model are not identical. Theoretically, this could lead to an over-approximation of knowledge; however, in practice this does not happen because no two zero-knowledge proofs of knowledge use the same randomness. Finally, we can only model simple relations between secret and public information; for instance, we cannot model a zero-knowledge proof with two public keys as public information, in which the prover proves that he knows at least one of the corresponding private keys.

We mention two final aspects that the above model does not take into account. First, from two runs of a Σ-protocol using the same prover randomness, the secret can be derived: in case of the Schnorr proof (Figure 6.4), by computing $(r - r')/(c - c')$ from transcripts (a, c, r) and (a, c', r'). This is a general property of Σ-protocols called *special soundness*. As mentioned above, we assume that randomness does not re-occur, so we can ignore this property. Second, an actor can "simulate" a Σ-protocol without knowing the secret information by first generating the challenge and response and from that determining the commitment. Such a simulation has the exact same form as a Σ-protocol, but because the randomness in the commitment is

unknown, it cannot be used to derive a secret corresponding to the public information. Because simulations cannot help to derive information from actually executed Σ-protocols, they are not relevant for knowledge analysis.

This extension appeared in earlier work[16]. We are not aware of other formalisations of zero-knowledge proofs of knowledge using formal methods, but there are models for *non-interactive* zero-knowledge proofs. Intuitively, in such proofs, the challenge from the verifier is replaced by a cryptographic hash of previous protocol messages so that the complete proof consists of just one message. As our model, existing formalisations of these non-interactive proofs only capture limited sets of properties that essentially uniquely relate the private and public information. Several works, e.g. by Backes et al.[17], Dong et al.[18], and Smyth et al.[19], have modelled non-interactive zero-knowledge proofs by an equational theory. These models, unlike ours, capture the relation between the secret and public information by means of a verification relation, making them more suitable for analysing properties in the presence of an attacker. Other than that, these models are similar to ours. Camenisch et al.[20] model system evolution due to correct non-interactive proofs. Their syntax for modelling proofs is similar to ours; but unlike us, they do not consider deriving knowledge from a proof.

[16] Veeningen et al. (2014)

[17] Backes et al. (2008)

[18] Dong et al. (2012)

[19] Smyth et al. (2012)

[20] Camenisch et al. (2010)

6.5 *Anonymous Credentials and Issuing*

We now propose a model of a particular anomyous credential scheme and its issuing protocol for the rule-based framework of Chapters 3 and 5. In general, a credential is an assertion, signed using the secret key of an *issuer*, of the link between the *owner*'s identifier and her identity attributes. An *anonymous credential* is a special type of credential that can be issued and shown without anybody but its owner learning the identifier. Hence, it can be used completely anonymously.

In particular, we model the scheme from Bangerter et al.[21] based on SRSA-CL signatures. Anonymous credentials are represented by function symbol $\mathsf{cred}/5$. Namely, $\mathsf{cred}(\mathsf{i}, \mathsf{k}^-, \mathsf{d}, \mathsf{n}_o, \mathsf{n}_i)$ is an anonymous credential with owner identifier i; issuer secret key k^-, owner attributes d, owner-contributed randomness n_o, and issuer-contributed randomness n_i. Anonymous credentials are issued in an issuing protocol, represented by function symbol $\mathsf{icred}/10$. Here, $\mathsf{icred}(\mathsf{i}, \mathsf{k}^-, \mathsf{d}, \mathsf{n}_1, \mathsf{n}_o, \mathsf{n}_3, \mathsf{n}_4, \mathsf{n}_i, \mathsf{n}_6, \mathsf{n}_7)$ represents a full transcript of the issuing protocol of an anonymous credential $\mathsf{cred}(\mathsf{i}, \mathsf{k}^-, \mathsf{d}, \mathsf{n}_o, \mathsf{n}_i)$, in which the owner additionally contributes randomness $\mathsf{n}_1, \mathsf{n}_3, \mathsf{n}_7$, and the issuer additionally contributes randomness $\mathsf{n}_4, \mathsf{n}_6$. Before the issuing protocol takes place, the owner is assumed to have sent a commitment to identifier i using randomness n_1, represented with function symbol $\mathsf{rc}/2$ as $\mathsf{rc}(\mathsf{i}, \mathsf{n}_1)$. Anonymous credentials are shown using zero-knowledge proofs (see Section 6.4).

[21] Bangerter et al. (2004)

The construction and elimination rules for cred, icred, and rc are

Construction/Elimination Rules $\mathsf{cred}(i,k^-,d,n_o,n_i)\leftarrow i,k^-,d,n_o,n_i \quad \mathsf{cred}(i,k^-,d,n_o,n_i)\xrightarrow{\doteq \mathsf{pk}(k^-),\doteq i,\doteq d}\{\mathsf{pk}(k^-),i,d\}$

$\mathsf{rc}(m,r)\leftarrow m,r \quad \mathsf{icred}(i,k^-,d,n_1,n_o,n_3,n_4,n_i,n_6,n_7)\leftarrow i,k^-,d,n_1,n_o,n_3,n_4,n_i,n_6,n_7$

$\mathsf{icred}(\ldots)\xrightarrow{\doteq n_o}\mathsf{cred}(i,k^-,d,n_o,n_i) \quad \mathsf{icred}(\ldots)\xrightarrow{\doteq n_3}\{i,n_1,n_o\} \quad \mathsf{icred}(\ldots)\xrightarrow{\doteq n_6}k^- \quad \mathsf{icred}(\ldots)\rightarrow\mathsf{pk}(k^-) \quad \mathsf{icred}(\ldots)\rightarrow\mathsf{rc}(i,n_1)$

$\mathsf{icred}(\ldots)\rightarrow d \quad \mathsf{icred}(\ldots)\xrightarrow{\doteq i,\doteq n_o}\{i,n_o\} \quad \mathsf{icred}(\ldots)\xrightarrow{\doteq \mathsf{cred}(i,k^-,d,n_o,n_i)}\mathsf{cred}(i,k^-,d,n_o,n_i)$

Transmission Validity $a(\mathsf{id}_a)\rightarrow b(\mathsf{id}_b) : \mathsf{icred}(\mathsf{i},\mathsf{k}^-,\mathsf{d},\mathsf{n}_1,\mathsf{n}_o,\mathsf{n}_3,\mathsf{n}_4,\mathsf{n}_i,\mathsf{n}_6,\mathsf{n}_7)$ valid if:

$\{\mathsf{id}_a,\mathsf{id}_b,\mathsf{pk}(\mathsf{k}^-),\mathsf{i},\mathsf{n}_1,\mathsf{n}_o,\mathsf{n}_3,\mathsf{n}_7\}$ determinable by a, $\{\mathsf{k}^-,\mathsf{d},\mathsf{n}_4,\mathsf{n}_i,\mathsf{n}_6\}$ determinable by b

Figure 6.5: Formal model of anonymous credentials and their issuing protocol: "…" is short for $i,k^-,d,n_1,n_o,n_3,n_4,n_i,n_6,n_7$ and $* \overset{*}{\rightarrow} \{z_i\}$ is short for the set of rules $\{* \overset{*}{\rightarrow} z_i\}$; reconstruction rules and testing rules from elimination are implicit

shown in Figure 6.5. From an anonymous credential cred, the public key, identifier, and attributes can together be tested. A commitment is modelled as function symbol rc without elimination rules (apart from reconstruction). From an issuing protocol icred, the owner can derive the credential using n_o (note that the issuer typically does not know n_o, so he cannot learn the credential), and his secret identifier and some randomness using n_3; the issuer can derive his secret key using n_6. The issuer public key, commitment, and owner attributes can be derived. Also, various combinations of items can be tested. To model the evolution of a system due to executing a credential issuing protocol, we adapt the formalism of Section 6.3. Namely, $a(\mathsf{id}_a)\rightarrow b(\mathsf{id}b) : \mathsf{icred}(\mathsf{i},\mathsf{k}^-,\mathsf{d},\mathsf{n}_1,\mathsf{n}_o,\mathsf{n}_3,\mathsf{n}_4,\mathsf{n}_i,\mathsf{n}_6,\mathsf{n}_7)$ represents a credential issuing protocol with owner a and issuer b. Validity of this transmission is shown in Figure 6.5. In particular, note that the owner contributes secret identifier i, while the issuer contributes private key k^- and attributes d.

As with our model of zero-knowledge proofs, we argue that this model is accurate assuming that the nonces from cred and icred do not occur in other messages in the knowledge base. More precisely, the owner and issuer *should* know their contributed randomness as context items; but apart from this, randomness n_o,n_3,n_4,n_i,n_6,n_7 should only occur in one icred message and its corresponding issued credential cred. (No such assumptions are made for n_1: in fact, it may be shared between multiple credential issuings.)

For examples of cred and icred, and of credential showing using zk, see the formalisation of Identity Mixer in Chapter 7.

We now argue the accuracy of our model of anonymous credentials by modelling how they are based on SRSA-CL signatures[22]. SRSA-CL signatures allow a signer to produce signatures on committed values. Formally, let srsacomm/3 denote a SRSA-CL commitment[23], where $\mathsf{srsacomm}(pk,i,na)$ denotes a commitment to the user's identifier i using the issuer's public key pk and randomness na. Function symbol srsacomm has standard construction rule $\mathsf{srsacomm}(pk,i,na)\leftarrow pk,i,na$ and no elimination rules. A SRSA-CL signature is modelled by function symbol srsas/5, where $\mathsf{srsas}(k^-,i,d,n_a,n_b)$ denotes a signature using private key k^- and randomness n_a,n_b on identifier i and list of attributes d.[24] SRSA-CL signatures can be constructed in the normal way from their parts,

[22] Camenisch and Lysyanskaya (2003); and Bangerter et al. (2004)

[23] Message C in Bangerter et al. (2004)

[24] Technically, different choices for n_a and n_b can lead to content equivalent signatures. However, this will not happen in practice assuming they are chosen at random.

$$a(\mathsf{id}_a) \to b(\mathsf{id}_b) \ : \mathsf{srsacomm}(\mathsf{pk}(\mathsf{k}^-), \mathsf{i}, \mathsf{n}_o);$$
$$a(\mathsf{id}_a) \to b(\mathsf{id}_b) \ : \mathsf{zk}(\{\mathsf{i}, \mathsf{n}_1, \mathsf{n}_o\}, \{\mathsf{pk}(\mathsf{k}^-), \mathsf{rc}(\mathsf{i}, \mathsf{n}_1), \mathsf{srsacomm}(\mathsf{pk}(\mathsf{k}^-), \mathsf{i}, \mathsf{n}_o)\}, \mathsf{n}_3, \mathsf{n}_4);$$
$$b(\mathsf{id}_b) \to a(\mathsf{id}_a) \ : \{\mathsf{srsas}(\mathsf{k}^-, \mathsf{i}, \mathsf{d}, \mathsf{n}_o, \mathsf{n}_i), \mathsf{n}_i\};$$
$$b(\mathsf{id}_b) \to a(\mathsf{id}_a) \ : \mathsf{zk}(\mathsf{k}^-, \{\mathsf{pk}(\mathsf{k}^-), \mathsf{srsacomm}(\mathsf{pk}(\mathsf{k}^-), \mathsf{i}, \mathsf{n}_o), \mathsf{d}, \mathsf{n}_i, \mathsf{srsas}(\mathsf{k}^-, \mathsf{i}, \mathsf{d}, \mathsf{n}_o, \mathsf{n}_i)\}, \mathsf{n}_6, \mathsf{n}_7)$$

Figure 6.6: Issuing protocol for anonymous credentials modelled in terms of SRSA-CL signatures

but also from a SRSA-CL commitment[25]:

$$\mathsf{srsas}(k^-, i, d, n_a, n_b) \leftarrow k^-, i, d, n_a, n_b$$
$$\mathsf{srsas}(k^-, i, d, n_a, n_b) \leftarrow \mathsf{srsacomm}(\mathsf{pk}(k^-), i, n_a), k^-, d, n_b$$

[25] Note that, formally speaking, the second construction rule is not allowed in our framework because variable k^- occurs several times. We ignore this formal problem in this discussion.

Moreover, SRSA-CL signatures admit signature verification and derivation of nonces:

$$\mathsf{srsas}(k^-, i, d, n_a, n_b) \to n_a \quad \mathsf{srsas}(k^-, i, d, n_a, n_b) \to n_b$$
$$\mathsf{srsas}(k^-, i, d, n_a, n_b) \xrightarrow{\doteq \mathsf{pk}(k^-), \doteq i, \doteq d} \{\mathsf{pk}(k^-), i, d\}.$$

An anonymous credential $\mathsf{cred}(\mathsf{i}, \mathsf{k}^-, \mathsf{d}, \mathsf{n}_o, \mathsf{n}_i)$ based on SRSA-CL signatures is simply a SRSA-CL signature $\mathsf{srsas}(\mathsf{k}^-, \mathsf{i}, \mathsf{d}, \mathsf{n}_o, \mathsf{n}_v)$ along with randomness $\mathsf{n}_o, \mathsf{n}_v$. From this (and because we assume non-re-use of nonces), the construction and elimination rules for cred in Figure 6.5 follow.

Execution of an anonymous credential issuing protocol

$$a(\mathsf{id}_a) \to b(\mathsf{id}_b) \ : \ \mathsf{icred}(\mathsf{i}, \mathsf{k}^-, \mathsf{d}, \mathsf{n}_1, \mathsf{n}_o, \mathsf{n}_3, \mathsf{n}_4, \mathsf{n}_i, \mathsf{n}_6, \mathsf{n}_7)$$

between owner a and issuer b can be modelled as a trace using primitives srsas and srsacomm (Figure 6.6). Beforehand, the owner is assumed to have sent commitment $\mathsf{rc}(\mathsf{i}, \mathsf{n}_1)$ to her identifier to the issuer. In the first two messages, the owner sends a commitment to the secret identifier to the issuer, and proves that it is formed correctly; that is, that the commitment indeed contains the same identifier as $\mathsf{rc}(\mathsf{i}, \mathsf{n}_1)$. The issuer uses the commitment to construct a SRSA-CL signature on the identifier and attributes, and sends the signature along with his randomness. At this point, the owner knows the signature and the two pieces of randomness used in it: these three components together form the anonymous credential. (Note that the issuer does not know n_o, so he does not have the complete credential.) In the last step, the issuer proves that $\mathsf{srsas}(\mathsf{k}^-, \mathsf{i}, \mathsf{d}, \mathsf{n}_o, \mathsf{n}_i)$ is valid; this step is technically needed to ensure the security of the signature[26].

[26] Bangerter et al. (2004)

We obtain the model for icred in Figure 6.5 by analysing what knowledge can be derived from the messages in Figure 6.6. From the first message, $\{\mathsf{pk}(\mathsf{k}^-), \mathsf{i}, \mathsf{n}_o\}$ can be tested. Because $\mathsf{pk}(\mathsf{k}^-)$ can be derived from other messages of the protocol, we get a testing rule for $\{\mathsf{i}, \mathsf{n}_o\}$. From message two, $\{\mathsf{pk}(\mathsf{k}^-), \mathsf{rc}(\mathsf{i}, \mathsf{n}_1), \mathsf{srsacomm}(\mathsf{pk}(\mathsf{k}^-), \mathsf{i}, \mathsf{n}_o)\}$ can be derived. Out of these messages, $\mathsf{srsacomm}(\mathsf{pk}(\mathsf{k}^-), \mathsf{i}, \mathsf{n}_o)$ was already covered above; we add elimination rules for $\mathsf{pk}(\mathsf{k}^-)$ and

$\mathsf{rc}(\mathsf{i}, \mathsf{n}_1)$.[27] Using n_3, $\{\mathsf{i}, \mathsf{n}_1, \mathsf{n}_o\}$ can be derived. From the third message, the credential $\mathsf{cred}(\mathsf{i}, \mathsf{k}^-, \mathsf{d}, \mathsf{n}_o, \mathsf{n}_i)$ can be tested and derived given n_o.[28] Finally, from the fourth message, attributes d can be derived; and n_6 can be used to derive k^-.[29]

Finally, consider the validity of the transmission

$$a(\mathsf{id}_a) \to b(\mathsf{id}_b) \ : \ \mathsf{icred}(\mathsf{i}, \mathsf{k}^-, \mathsf{d}, \mathsf{n}_1, \mathsf{n}_o, \mathsf{n}_3, \mathsf{n}_4, \mathsf{n}_i, \mathsf{n}_6, \mathsf{n}_7).$$

Assuming fresh nonces, determinability by a of $\{\mathsf{id}_a, \mathsf{id}_b, \mathsf{pk}(\mathsf{k}^-), \mathsf{i}, \mathsf{n}_o\}$ is required for the first transmission in Figure 6.6. For the first ZK proof, determinability by a of n_1 and n_3 is required; and determinability by b of n_4. The next message means determinability by b of $\{\mathsf{k}^-, \mathsf{d}, \mathsf{n}_i\}$. The last ZK proof additionally means determinability by b of $\{\mathsf{pk}(\mathsf{k}^-), \mathsf{n}_6\}$, and by a of n_7. We get the result shown in Figure 6.5.[30]

[27] Also, n_3 can be tested, but this is not relevant as we assume it does not occur outside of the protocol.

[28] Also, the signature can be verified to obtain a testing rule for $\{\mathsf{pk}(\mathsf{k}^-), \mathsf{i}, \mathsf{d}, \mathsf{n}_o\}$, but other rules already cover this. Moreover, n_v can be derived; again, this is not relevant because it is assumed not to occur outside of the credential and issuing protocol.

[29] As before, we can test n_6 but this is not relevant assuming non-reoccurence.

[30] Technically, a does not need d to run the protocol, and b does not need $\mathsf{rc}(\mathsf{i}, \mathsf{n}_1)$; however, in practice, they will use this information to check whether the data supplied matches their expectations.

Discussion

This extension appeared in earlier work[31]. Recently, several other works have proposed formal models of anonymous credentials. Li et al.[32] model anonymous credentials, but only consider operational aspects, i.e. they model how the protocol takes place but not what knowledge can be obtained from observing it. Camenisch et al.[33] model credentials and their showing protocol. The model of credentials is similar to ours, and it includes a rule to obtain a credential from a committed message as our low-level formalisation above. Like us, they formalise the showing protocol in terms of ZK proofs. Credential issuing is not considered. Finally, Smyth et al.[34] model joining and signing protocols for ECC-based Direct Anonymous Attestation, which are very similar to issuing and showing protocols for a variant of the scheme we have presented here. Although our model is based on a different signature scheme[35] and specified at a higher level, their model of signatures generally corresponds to our low-level model of signatures from committed messages.

[31] Veeningen et al. (2014)

[32] Li et al. (2009)

[33] Camenisch et al. (2010)

[34] Smyth et al. (2012)

[35] We model SRSA-CL credentials by Camenisch and Lysyanskaya (2003); the variant is based on BM-CL signatures by Camenisch and Lysyanskaya (2004)

7
Comparing Identity Management Systems

Contents

IN THIS CHAPTER, we demonstrate our framework by comparing a number of identity management (IdM) systems. IdM systems[1] offer reliable on-line identification and authentication of users to service providers by outsourcing these tasks to "identity providers". Identity providers endorse information about their users, and provide means for authenticating a user in a service provision. To organisations, identity providers offer reduced cost for obtaining reliable user information; to users, they offer increased convenience by letting them reuse authentication credentials. The amount of personal information exchanged in such systems makes privacy a critical issue; this is reflected by the large number of privacy-enhancing IdM systems that have been proposed[2]. However, while high-level sketches of privacy issues[3] and comparisons of systems[4] exist, no comprehensive set of relevant privacy properties for IdM systems has been proposed, nor do there exist precise formal comparisons. We demonstrate that our framework can be used to perform such a comparison.

[1] E.g., Sommer et al. (2008), Kellomäki (ed.) (2009), Erdos and Cantor (eds.) (2005)

[2] E.g., Bangerter et al. (2004), Chadwick and Inman (2009), Vossaert et al. (2011)

[3] E.g., Alpár et al. (2011), Bhargav-Spantzel et al. (2007b), Hansen et al. (2004), Landau et al. (2009)

[4] E.g.,Independent Centre for Privacy Protection Schleswig-Holstein (2003), Hoepman et al. (2008)

To perform the privacy comparison, we perform the four steps

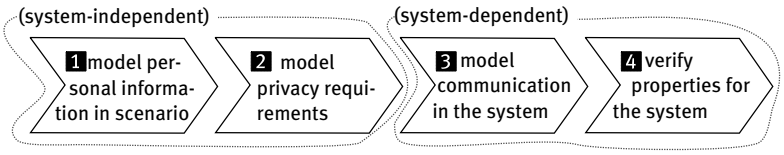


Figure 7.1: Steps of a privacy analysis using our framework

shown from Figure 7.1. The *first step* is to model all personal information using a PI Model (see Section 2.1), and to model the initial knowledge of each actor as a view on that PI Model (see Section 2.2). As noted before, this not only means modelling the personal information as used in protocol instances; but also modelling other knowledge of personal information. This way, we can assess how links can be established between knowledge learned from protocol instances and other knowledge.

The *second step* is to model privacy properties, i.e., which personal information should become known or remain unknown to which actors in the system (see Section 2.3). These properties are phrased in terms of the views of actors after communication has taken place. These first two steps are performed independently from the particular systems to be analysed.

The *third step* is to model the exchange of information in the information systems. We model this exchange using traces (see Section 6.3), leading to a state whose knowledge bases can be analysed using our rule-based model (see Chapter 3).

The *fourth step* is to verify which systems satisfy which properties. This step is performed automatically using our Prolog tool for the formal analysis of privacy in communication protocols (Section 3.7)[5]. Given a PI Model, a set of formalised properties, an initial state and a trace, the tool first determines the state of the system after communication; then computes the corresponding views of the actors in the system, and finally determines which properties hold in these views.

[5] The tool and the formal models of identity management systems presented in this chapter are available at `http://code.google.com/p/objective-privacy/`

Outline In this chapter:

- We introduce identity management and its privacy issues (§7.1);
- We propose a comprehensive set of privacy properties of identity management systems (§7.2);
- We present four identity management systems whose privacy properties we will compare (§7.3);
- We perform the formal privacy analysis according to the steps described above, and analyse the results (§7.4–§7.7);
- We show how the above, instantiated, analysis can be generalised using the symbolic model of Chapter 5 (§7.8);
- We finish by discussing relevant literature on privacy in identity management (§7.9).

7.1 Identity Management

As providers of on-line services are offering more and more customisation to their users, they need to collect more and more of their personal information. Traditionally, each service provider would manage the accounts of users separately. However, this identity

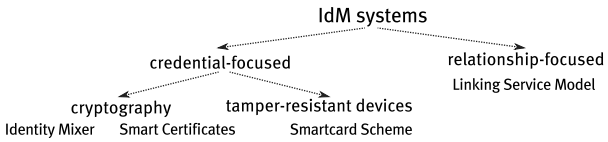


Figure 7.2: Taxonomy of IdM systems

management model, called the *isolated user identity management model*[6], has disadvantages for both users and service providers: the user has to manually provide and update her information and keep authentication tokens for each service provider, whereas it is hard for the service provider to obtain guarantees that the information given by the user is correct.

[6] Jøsang and Pope (2005)

This problem is commonly addressed using an *Identity Management (IdM) System*. Intuitively, the task of managing and endorsing identity information is delegated to *identity providers*. Identity management is split up in two phases: *registration* and *service provision*. At registration, users establish accounts at (possibly multiple) identity providers.[7] Service provision is the phase when a user requests a service from a service provider: at this point, user attributes required for the service provision need to be collected and sent to the service provider.

[7] This includes *identification*: i.e., the user transfers her attributes to the identity provider, and the identity provider possibly checks them. However, both the transfer and checking of attributes performed by the identity provider are out of the scope of this chapter.

Bhargav-Spantzel et al.[8] divide IdM systems into two main categories depending on whether or not the identity providers are involved in the service provision phase: *credential-focused* and *relationship-focused* systems (also called network-based and claim-based systems by Alpár et al.[9]). Figure 7.2 shows a taxonomy of IdM systems.

[8] Bhargav-Spantzel et al. (2007a)

[9] Alpár et al. (2011)

In credential-focused IdM systems, the user gets long-term credentials from the identity provider in the registration phase, that she can directly present to the service providers in the service provision phase. These credentials contain her identity attributes. We distinguish between two mechanisms employed to prevent the user from tampering with them, namely *cryptography* and *tamper-resistant devices*. Credential-focused systems relying on cryptography include CardSpace[10], U-Prove[11] and Identity Mixer[12]. The system presented by Vossaert et al.[13] relies on the use of a smartcard as a tamper-resistant device.

[10] Nanda (2007)

[11] Brown et al. (2010)

[12] Bangerter et al. (2004)

[13] Vossaert et al. (2011)

In relationship-focused IdM systems, in contrast, identity providers present the attributes to service providers. During the registration phase, identity providers establish shared identifiers to refer to each other's identity of the user. During the service provision phase, the user authenticates to an identity provider. The identity provider then sends attributes to the service provider (possibly indirectly via the user). If needed, the shared identifiers established during registration are used to collect (or *aggregate*[14]) attributes held by other identity providers without the user having to authenticate to them as well. The combination of reliance on authentication performed by another party and exchange of identity information is sometimes referred to as *federated identity manage-*

[14] Chadwick and Inman (2009)

ment[15,16]. Relationship-focused systems include Liberty Alliance[17], Shibboleth[18], and the linking service model[19].

Because in IdM systems, large amounts of personal information are processed by many different parties, privacy has become a major concern[20]. In such systems, privacy threats posed by authorised insiders are nowadays considered to be a critical problem besides outsider attacks on cryptographic protocols[21]. Insiders may compile comprehensive user profiles to sell or use for secondary purposes such as marketing. These profiles can include sensitive information that is explicitly transferred by the user, but also information that is transferred *implicitly*[22]. For instance, the mere fact that a user performed a transaction at a certain service provider may be privacy-sensitive. In addition, profiles held by different parties may be combined[23] to compile even more comprehensive profiles. *Privacy-enhancing IdM systems*[24] aim to minimise the amount of information disclosed as well as prevent that different pieces of information can be linked together[25].

[15] Jøsang and Pope (2005); and Smedinghoff (2009)

[16] Note that this term is also used to describe the general concept of sharing information between different domains (cf. Alpár et al. (2011)) or the mere use of multiple identity providers (cf. Independent Centre for Privacy Protection Schleswig-Holstein (2003)). To avoid confusion, we will not use it further.

[17] Hodges et al. (2006)

[18] Erdos and Cantor (eds.) (2005)

[19] Chadwick and Inman (2009)

[20] Hansen et al. (2004); and Spiekermann and Cranor (2009)

[21] Fyffe (2008)

[22] Spiekermann and Cranor (2009)

[23] Spiekermann and Cranor (2009)

[24] E.g., Bangerter et al. (2004), Chadwick and Inman (2009), Vossaert et al. (2011)

[25] Hansen et al. (2004)

7.2 Privacy Properties

We now present a set of privacy properties for IdM systems. We have obtained these properties by analysing the information that actors can learn; considering which knowledge should be avoided; and systematically grouping this knowledge into properties according to what kind of knowledge it is, and who should or should not learn it. We validate our set of properties in two different ways. First, we check if they cover relevant privacy properties discussed in the literature. For this, we have studied taxonomies of privacy in identity management[26] and the proposals for the identity management systems analysed in this chapter[27], and verified if all properties discussed in these works are covered by our properties. Second, we check if they cover all possible situations expressible in our model that can lead to privacy risks. For this, we have systematically considered all elementary detectability, linkability and involvement properties expressible in our model, checked which of these can lead to privacy risks, and verified that the relevant ones are covered by our properties.

[26] Bhargav-Spantzel et al. (2007b); and Hansen et al. (2004)

[27] Bangerter et al. (2004), Chadwick and Inman (2009), Vossaert et al. (2011)

Table 7.2 lists our privacy properties, also showing in which existing works they are discussed. We first present our properties, then compare them to the aforementioned literature. Coverage of situations expressible in our model is discussed in Section 7.5.

The basic *functional requirement* for IdM systems is that the service provider learns the attributes it needs[28]: *attribute exchange* (AX). Note that in one service provision, a service provider may need attributes from several identity providers.

[28] Bhargav-Spantzel et al. (2007b)

Privacy properties cover that certain personal information should not be learned by certain actors. Privacy-enhancing systems should minimise the amount of information learned, and the extent to which it can be linked together[29]. The first aspect, information

[29] Hansen et al. (2004)

Functional requirements	**Description**	**References**
Attribute exchange (AX)	The service provider learns the value of the required attributes/predicates of the user requesting the service.	1,2,3,5,6
Privacy properties	**Description**	
Irrelevant attribute undetectability (SID)	The service provider does not learn anything about attribute values irrelevant to the transaction.	1,2,5,6
Predicate-attribute undetectability (SPD)	The service provider does not learn anything about attributes apart from the predicates he needs to know.	1,2,5,6
IdP attribute undetectability (ID)	Identity providers do not learn anything about the user's attributes from other identity providers.	-
Mutual IdP involvement undetectability (IM)	One identity provider does not learn whether a given user also has an account at another identity provider.	3
IdP-SP involvement undetectability (ISM)	Identity providers do not learn which service providers a user uses.	-
Session unlinkability (SL)	A service provider cannot link different sessions of the same user.	1,2,3,4,6
IdP service access unlinkability (IL)	Identity providers cannot link service access to the user profile they manage.	4
IdP profile unlinkability (IIL)	Collaborating identity providers cannot link user profiles.	4,6
IdP-SP unlinkability (ISL)	Identity providers and service provider cannot link service accesses to user profiles at the identity provider.	1,4,6
Accountability properties	**Description**	
Anonymity revocation (AR)	Service provider and identity providers (possibly with help from trusted third party) can reconstruct the link between service access and user profile.	1,2,4,6

Table 7.2: Properties for IdM systems and literature in which they are discussed (1: Bangerter et al. (2004), 2: Bhargav-Spantzel et al. (2007b), 3: Chadwick and Inman (2009), 4: Hansen et al. (2004), 5: Park and Sandhu (1999), 6: Vossaert et al. (2011))

learned, can be further divided into explicitly and implicitly transferred information[30]. *Detectability* properties capture explicitly transferred information: information about the user's attributes. *Involvement* properties capture information about whether actors know about each other's involvement with the user: a kind of implicitly transferred personal information. The second aspect, information linked together, is captured by *linkability* properties: namely, properties capturing that (combinations of) parties should not be able to link personal information from different sessions, databases, etc.

[30] Spiekermann and Cranor (2009)

We define three detectability properties. The first two are about the service provider learning no more than strictly necessary: no attributes that he does not need to know (*irrelevant attribute undetectability*, SID), and no complete attribute values if all he needs to know is whether or not an attribute satisfies a certain predicate[31] (*predicate-attribute undetectability*, SPD). These properties limit the user profile a service provider can construct. In addition, IdM systems should guarantee that identity providers do not learn any value or predicate of attributes stored at other identity providers: we call this property *IdP attribute undetectability* (ID).

[31] Bangerter et al. (2004)

Involvement properties address the fact that the mere interaction of a user with certain identity or service providers implies a business relation which can be privacy-sensitive. For instance, ownership

of credentials can be sensitive[32] in domains such as healthcare, insurance, or finance. In addition, even if individual credentials are not sensitive, the precise combination of credentials held by a user may help identify her[33]. It is natural in identity management that the service provider learns which identity providers certify the user's attributes: this allows him to judge their correctness. However, identity providers should not know the identity of other identity providers the user has an account at[34]: we define this as *mutual IdP involvement undetectability* (IM). In the same way, a user might want to keep hidden from her identity providers the fact that she interacts with a certain service provider: we call this property *IdP-SP involvement undetectability* (ISM).

[32] Seamons et al. (2003)

[33] Pashalidis and Meyer (2006)

[34] Chadwick and Inman (2009)

Linkability is another fundamental privacy concern because it determines what user profiles can be constructed from the data that is collected[35]. To prevent a service provider from accumulating (behavioural) information, an IdM system should ensure it cannot link different service provisions to the same user: *session unlinkability* (SL). Indeed, in many cases the service provider does not need to know the identity of the user: for instance, if a user wishes to read an on-line article, the only information that is required is that she has a valid subscription.

[35] Pfitzmann and Hansen (2009)

Another concern is that parties can build more comprehensive user profiles by sharing their personal information. To prevent this, they should not know which profiles are about the same user[36]. A very strong privacy guarantee in this vein is that identity providers and service providers cannot link service provisions to the user: *IdP-SP unlinkability* (ISL). *IdP profile unlinkability* (IIL) is a weaker privacy guarantee requiring that two collaborating identity providers (without help from the service provider) cannot link their profiles. *IdP service access unlinkability* (IL) is about the link between a service provision and the user profile at an identity provider, thus measuring whether identity providers are aware of individual service provisions.

[36] Hansen et al. (2004)

An *accountability property* counterbalances the privacy guaranteed by the ISL property. Namely, if the user misbehaves, it should be possible to identify her[37]. Several IdM systems[38] introduce a trusted third party that, in such cases, can help with the identification. The *anonymity revocation* (AR) property states that, possibly with the help of this trusted third party, the service provider and identity providers are able to revoke the anonymity of a transaction. (Note that in particular, AR also holds if the service provider and identity providers can revoke anonymity without needing the trusted third party.)

[37] Bangerter et al. (2004)

[38] E.g., Bangerter et al. (2004), Vossaert et al. (2011)

When comparing our properties to those found in existing taxonomies[39], we find that our properties are generally more detailed. Bhargav-Spantzel et al.[40] present three properties on data minimisation: conditional release, selective disclosure, and unlinkability. These three properties correspond to anonymity revocation and IdP-SP unlinkability; irrelevant attribute and predicate-attribute

[39] Bhargav-Spantzel et al. (2007b); and Hansen et al. (2004)

[40] Bhargav-Spantzel et al. (2007b)

undetectability; and session unlinkability, respectively (for selective disclosure, the authors do not distinguish between attributes and predicates). The authors also mention policy support, which we do not cover. On the other hand, our other properties are not addressed. Hansen et al.[41] mention "user-controlled linkage of personal data" as the underlying principle of privacy-enhancing identity management. This includes our unlinkability properties (although Hansen et al. do not identify them separately), but also a "control" aspect of privacy which we do not cover. Hansen et al. also stress that the desired degree of linkability depends on the application, mentioning revocation in particular.

[41] Hansen et al. (2004)

As shown in the table, many of our properties are discussed by designers of IdM systems[42]. We compare our properties to those claimed by designers (including the ones we do not cover) when discussing these systems in Section 7.3.

[42] Bangerter et al. (2004); Chadwick and Inman (2009); and Vossaert et al. (2011)

7.3 Four Systems

We now present the four IdM systems we have formally analysed. We consider one traditional system, *smart certificates* by Park and Sandhu[43], for whose development privacy was not a primary concern; it can be classified as credential-focused and relying on cryptography. We then consider three systems designed with privacy in mind: the *linking service model* by Chadwick and Inman[44], a relationship-focused IdM system; *Identity Mixer* by Bangerter et al.[45], a credential-focused system relying on cryptographic protocols; and a credential-focused IdM system based on smartcards by Vossaert et al.[46] we will refer to as the *Smartcard scheme*.

[43] Park and Sandhu (1999)

[44] Chadwick and Inman (2009)

[45] Bangerter et al. (2004)

[46] Vossaert et al. (2011)

For our analysis, we aim to cover differ kinds of IdM systems that exist in the literature. In particular, this means selecting credential-focused and relationship-focused systems[47]. For the former type, Identity Mixer has received a lot of attention in the research community. For the latter type, the linking service is one of few proposals supporting multiple identity providers that takes privacy into account[48]. We also include the smartcard scheme because it is a recent proposal in a completely different direction than the previous two. Of course, our formal results are about these particular systems; however, when analysing the results, we will also discuss to what extent they generalise to similar systems.

[47] Cf. Alpár et al. (2011), Bhargav-Spantzel et al. (2007b)

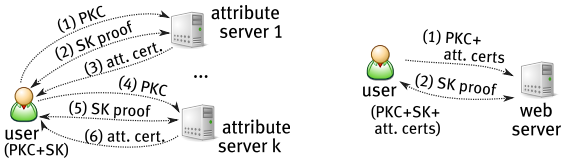
[48] Chadwick and Inman (2009)

We now briefly discuss these systems and the privacy guarantees that they have been designed to provide.

Smart Certificates

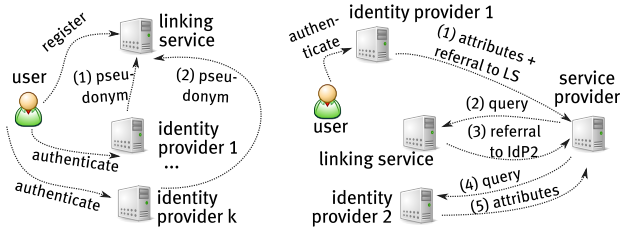
Park and Sandhu[49] propose an IdM system built on top of a Public Key Infrastructure (PKI). In a PKI, a certificate authority (CA) issues certificates stating that a certain public key belongs to a certain user. A user authenticates by proving knowledge of the secret key corresponding to this public key. Identity providers issue certificates

[49] Park and Sandhu (1999)



(a) Registration phase (b) Service provision phase

Figure 7.3: Smart certificates



(a) Registration phase (b) Service provision phase

Figure 7.4: Linking service model

that link attributes to the public key certificate. In our analysis, we consider one particular variant described by Park and Sandhu: the user-pull model with long-lived certificates obtained during registration.

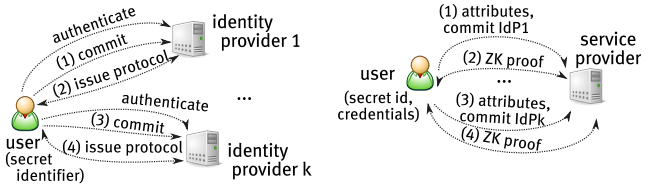
The flow of information is summarised in Figure 7.3. In the registration phase (Figure 7.3(a)), the user gets an attribute certificate from an identity provider (called "attribute server" in Park and Sandhu (1999)), which enables her to present her attributes to others. This involves three steps: (1) the user presents her public key certificate; (2) she proves that she also knows the corresponding secret key (this is an interactive protocol shown as a two-sided arrow in the figure); and (3) the attribute server issues an attribute certificate. The process is then repeated with other identity providers (steps (4) to (6)). The attributes in the certificate are signed using the attribute server's secret key and hence cannot be tampered with by the user. During service provision (Figure 7.3(b)), the user exchanges attributes with the service provider ("web server") in two steps: (1) she presents her public key certificate and the attribute certificates containing the attributes needed; and (2) she proves knowledge of the corresponding secret key.

The system presented in Park and Sandhu (1999) is mainly designed to satisfy the attribute exchange (AX) property in a secure way ("the attributes of individual users are provided securely"). Privacy concerns are addressed in an extension of the system in which some attributes in a credential are encrypted in such a way that they can only be read by an "appropriate" server, corresponding to our SID/SPD properties. However, we will consider the original scheme in which SID/SPD are not claimed to hold.

Linking Service Model

The linking service model[50] is a relationship-focused IdM system. Its main goal is to facilitate the collection of user attributes from different identity providers in a privacy-friendly way without the user having to authenticate to each identity provider separately. To

[50] Chadwick and Inman (2009)



(a) Registration phase (b) Service provision phase

Figure 7.5: Identity Mixer

this end, this model includes a *linking service* which is responsible for holding the links between profiles of the user at the different identity providers without knowing any personal information about the user.

The flow of information is summarised in Figure 7.4. During registration (Figure 7.4(a)), the user first creates an anonymous account at the linking service LS. LS requests the identity providers to authenticate the user; each identity provider generates a pseudonym for the user and sends it to LS (steps (1) and (2)). (The specific method of authentication between the user and the identity providers and linking service is out of our scope.) In the service provision phase (Figure 7.4(b)), the user authenticates to one particular identity provider IdP_1. IdP_1 provides the service provider SP with an "authentication assertion" containing the attributes requested from it, and a referral to LS (1). The referral is an encryption of the pseudonym shared between IdP_1 and LS that only LS can decrypt. SP sends this referral to LS (2), which responds by sending a similar referral to other identity providers (3). Finally, SP requests (4) and obtains (5) the required attributes from the other identity providers (for simplicity, we just show one other identity provider in the figure).

The linking service model aims to satisfy the attribute exchange property (AX) as well as a number of privacy properties[51]. In particular, the main goal of the linking service model is to guarantee that identity providers do not know the involvement of other identity providers (IM). Moreover, the model aims to achieve session unlinkability (SL) through the use of random user identifiers. Finally, the linking service should not learn the partial identities of the user for the service providers; that is, it does not learn any personal information about the user. We call this property *LS attribute undetectability* (LD); it is not listed in Table 7.2 because it is only relevant for this system; however, our analysis will include the verification of this property.

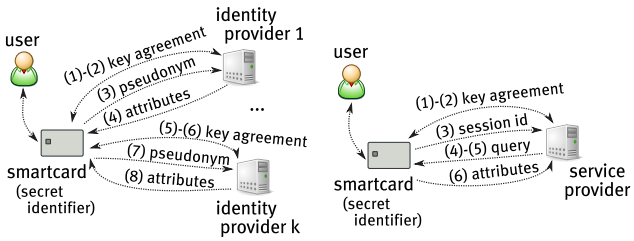
[51] Chadwick and Inman (2009)

7

Identity Mixer

Identity Mixer[52] is a credential-focused IdM system using a cryptographic primitive called anonymous credentials. These credentials link attributes to a user identifier, but are issued by identity providers and shown to service providers using protocols ensuring that neither party learns that identifier. Thus, nobody but the user knows whether different issuing or showing protocols were performed by

[52] Bangerter et al. (2004)



(a) Registration phase (b) Service provision phase

Figure 7.6: Smartcard scheme

the same user, while integrity of the attributes is still assured.

Figure 7.5 shows the information flows in Identity Mixer. During registration (Figure 7.5(a)), the user first sends a commitment to her (secret) identifier to an identity provider IdP_1 (1), after which the user and IdP_1 together run the credential issuing protocol (2). From this, the user obtains a credential with her attributes linked to her secret identifier, without IdP_1 learning the identifier. Communication with other identity providers is analogous (steps (3) and (4)). In the service provision phase (Figure 7.5(b)), the user shows information from several credentials to the service provider SP. She first shows her credential from one identity provider. To this end, she sends a message containing the attributes she wants to reveal, and "commitments" to the secret identifier and all other attributes (1). Next, she performs a zero-knowledge proof (2) which proves to SP that the attributes and commitments come from a valid credential issued by the identity provider, while revealing nothing else about the credential. Credentials issued by other identity providers are shown in the same way (steps (3) and (4)).

Identity Mixer is designed to satisfy a number of privacy properties[53]. In particular, it aims to satisfy both session unlinkability and IdP/SP unlinkability (together called "multi-show unlinkability" in Bangerter et al. (2004)) and irrelevant attribute and predicate-attribute undetectability (together called "selective show of data items" in Bangerter et al. (2004)). The system allows for providing the service provider with an encryption of some attributes for a trusted third party ("conditional showing of data items" in Bangerter et al. (2004)) that can be used for anonymity revocation. Apart from the data minimisation properties we defined, the system additionally allows credential issuing where an identity provider copies attributes from another certificate without knowing their values ("blind certification" in Bangerter et al. (2004)). The main motivation for this functionality comes from the use of these certificates for e-cash. In traditional identity management scenarios, such as ours, identity providers should know the attributes they endorse, so we do not consider this property in this work.

[53] Bangerter et al. (2004)

7

Smartcard Scheme

Vossaert et al.[54] propose a credential-focused IdM system which relies on a PKI for authentication and on smartcards (or other tamper-resistant devices) to ensure that attributes are not modified and

[54] Vossaert et al. (2011)

Scheme	AX	AR	SID	SPD	ID	IM	ISM	SL	IL	IIL	ISL
Smart certificates	✓										
Linking service	✓					✓		✓			
Identity Mixer	✓	✓	✓	✓				✓			✓
Smartcard scheme	✓	✓	✓	✓				✓		✓	✓

Table 7.3: Comparison of privacy properties claimed by the various systems

observed during their transmission from the identity provider to the service provider. Identity providers and service providers only communicate via the smartcard, and each has a different pseudonym of the user based on a secret user identifier stored on the smartcard.

The information flow in the scheme is shown in Figure 7.6. In the registration phase (Figure 7.6(a)), the smartcard SC and the first identity provider IdP_1 establish a secure, authenticated channel using a key agreement protocol (steps (1) and (2)). Over this secure channel, SC sends a pseudonym based on its secret identifier specific for IdP_1 (3); IdP_1 sends its attributes (4). Registration at other identity providers is similar (steps (5) to (8)). In a service provision (Figure 7.6(b)), SC and service provider SP establish a secure, authenticated channel as in the registration phase (steps (1) and (2)). SC generates a random session identifier (3); SP then specifies what attributes it wants, and how long they may have been cached (steps (4) and (5)). SC responds by giving the requested attributes. For anonymity revocation purposes, this response also includes Alice's identifier encrypted for the trusted third party (6).

The system is designed to meet several properties related to the knowledge of personal information[55]. The properties specified correspond to our notions of attribute exchange, session unlinkability, and anonymity revocation. Irrelevant attribute undetectability and predicate-attribute undetectability follow from their more general notion of "restricting released personal data". The Smartcard scheme also aims to fulfil IdP profile unlinkability and IdP/SP unlinkability by preventing collusion of identity and service providers.

[55] Vossaert et al. (2011)

Privacy Properties Claimed by Systems

Table 7.3 summarises the privacy claims for the systems. One goal of our formal analysis will be to verify whether these claims actually hold. In addition, we will analyse the systems against the complete range of identified properties in order to achieve a comprehensive comparison of their privacy features.

7.4 Step 1: Model Personal Information

Step 1 of our formal privacy comparison is to model personal information in a scenario. The scenario needs to be designed in such a way that all privacy properties to be verified (i.e., in this case, the ones in Table 7.2) can be phrased in terms of personal information occurring in the scenario. Thus, we include attributes that should

be disclosed (for AX), should not be disclosed (for SPD), and about which only a predicate should be disclosed (for SID); and we consider multiple identity providers (for IM, IL, and IIL) and sessions (for SL). Given these constraints, we design a scenario with realistic data.

In particular, we consider a scenario with four main actors: a user: Alice, a 65 year-old woman; a service provider: an e-book store; and two identity providers: one for Alice's address (the address provider) and one for Alice's subscription at some society (the subscription provider). In the *registration phase* of this scenario, Alice creates an account at both identity providers. The address provider stores three identity attributes of the user: the street, city, and age. The subscription provider stores two user attributes: date of subscription and subscription type.

In the *service provision phase*, Alice purchases books from the e-book store on two separate occasions. To this end, she needs to provide her personal information, endorsed by the identity providers, to the e-book store. The e-book store, for statistical purposes, demands to know the city that Alice comes from. Moreover, the store offers a discount to customers that are over 60 years old. As Alice is 65 years old, she is eligible for the discount. The e-book store, however, does not necessarily need to learn her exact birth date or age; Alice can just prove that she is over 60 years old. Moreover, the e-book store does not need to know that the purchases are both made by the same user. On the other hand, in case of abuse, the service provider does want to be able to link the purchase to Alice's profile at the address provider with the help of a trusted third party.[56]

[56] Note that the scenario does not cover the separate issue of anonymous payment of the e-book.

7

Our formalisation of this scenario as (views on) a PI Model is shown schematically in Figure 7.7. Figure 7.7(a) lists the actors in the system. The trusted third party ttp is included because of the anonymity revocation property; however, note that it only occurs in the Identity Mixer and Smartcard schemes.

Figures 7.7(b) and 7.7(c) summarise the contexts we use to model different representations of information about Alice and the other actors. Figure 7.7(b) lists all domains. The "$\cdot$" domain contains publicly known identifiers for the identity and service providers, and their private keys. The ι, κ, and λ domains represent databases of user information held by the respective parties. The π, η, ζ, and ξ domains represent the communication protocols that are executed during the scenario.[57] Figure 7.7(c) shows the profiles representing the actors in the different domains. For instance, in the $\cdot$, ι, κ and λ domains, Alice represented by the *al* profile; in the π, η, ζ, and ξ domains, she is represented by u. By naming these profiles differently, we emphasise that actors learn the information not as information about Alice, but as information about "the purchaser in transaction x", etc.

[57] For simplicity, all communication related to one service provision is modelled in a single domain. This expresses that parties involved in service provision without communicating directly (e.g., the linking service and IdP_2 in the linking service model) are able to link their views of the protocol. Alternatively, each pair of communication partners could have a separate domain.

Figures 7.7(d) and 7.7(e) define the pieces of personal information in the scenario, and the knowledge about them that actors hold in

$a \in \mathcal{A}$	Actor
al	Alice
ii	Address provider
is	Subscription provider
bs	E-book store
ttp	Trusted third party

(a) Actors

Dom.	Description
$\cdot$	identifiers/keys
ι	Alice's knowledge
κ	ii's user database
μ	is's user database
π	registration at ii
η	registration at is
ζ,ξ	service provisions

(b) Domains

Actor	Domains $\cdot$,ι,κ,μ	Domains π,η,κ,μ
al	al	u
ii	ii	$idp1$
is	is	$idp2$
bs	bs	sp
ttp	ttp	ttp

(c) Profiles

$$\{ip|^{\cdot}_{ii}, k^{-}|^{\cdot}_{ii}, ip|^{\cdot}_{is}, k^{-}|^{\cdot}_{is}, ip|^{\cdot}_{bs}, k^{-}|^{\cdot}_{bs}, k^{-}|^{\cdot}_{ttp}\}$$

(d) Information about other actors (anybody knows identifiers and public keys; actor knows own private key)

Info	al	ii	is	bs	Description
i	$\{i\|^{\iota}_{al}, i\|^{\zeta}_{u}, i\|^{\xi}_{u}\}$	-	-	-	Private identifier
i_{ii}	$\{i_{ii}\|^{\iota}_{al}, i_{ii}\|^{\pi}_{u}\}$	$\{i_{ii}\|^{\kappa}_{al}, i_{ii}\|^{\pi}_{u}\}$	-	-	Identifier at ii
d_1	$d_1\|^{\iota}_{al}$	$d_1\|^{\kappa}_{al}$	-	-	City
d_2	$d_2\|^{\iota}_{al}$	$d_2\|^{\kappa}_{al}$	-	-	Age
$d_2?_{gt60}$	$d_2?_{gt60}\|^{\iota}_{al}$	$d_2?_{gt60}\|^{\kappa}_{al}$	-	-	Age "> 60" predicate
d_3	$d_3\|^{\iota}_{al}$	$d_3\|^{\kappa}_{al}$	-	-	Address
i_{is}	$\{i_{is}\|^{\iota}_{al}, i_{is}\|^{\eta}_{u}\}$	-	$\{i_{is}\|^{\mu}_{al}, i_{is}\|^{\eta}_{u}\}$	-	Identifier at is
d_5	$d_5\|^{\iota}_{al}$	-	$d_5\|^{\mu}_{al}$	-	Subscription date
d_6	$d_6\|^{\iota}_{al}$	-	$d_6\|^{\mu}_{al}$	-	Subscription type
$\mathbf{d_7}$	-	-	-	$\{\mathbf{d_7}\|^{\zeta}_{u}, \mathbf{d_7}\|^{\xi}_{u}\}$	Transaction details
$\mathbf{ip}$	$\{\mathbf{ip}\|^{\pi}_{u}, \mathbf{ip}\|^{\eta}_{u}, \mathbf{ip}\|^{\zeta}_{u}, \mathbf{ip}\|^{\xi}_{u}\}$	-	-	-	IP address

(e) Information about Alice, and initial knowledge about this information held by actors

Figure 7.7: Schematic representation of PI Model and initial knowledge

the initial state. For simplicity, we give an explicit context-layer representation, and use notational conventions to implicitly describe the information and contents layers. Namely, when context items about the same actor using the same variable are denoted in the normal font (e.g. $i_{ii}|^{\pi}_{u}$ and $i_{ii}|^{\kappa}_{al}$), they represent the same information-layer item; when denoted in boldface (e.g. $\mathbf{ip}|^{\pi}_{u}$, $\mathbf{ip}|^{\eta}_{u}$), they all represent different information-layer items. Items of the form $i|^{*}_{*}$, $i_{*}|^{*}_{*}$, $k^{-}|^{*}_{*}$, and $ip|^{*}_{*}$ (for any $*$) are identifiers; items $d_{*}|^{*}_{*}$ are data items; other items are non-personal information. All representations of a single piece of information use the same variable. Because this scenario includes only one data subject, all pieces of information have unique contents, i.e., the information and contents layers coincide. We have one predicate $gt60$ representing if an age is over 60 (see Section 6.2). For instance, $d_2?_{gt60}$ represents the fact that the data item d_2 represents an age over 60.

Figure 7.7(d) defines the information available about ii, is, and bs. This information consists of a private key for each of the actors, and an identifier for ii, is, and bs. All actors know each other's identifiers and the public keys $\mathsf{pk}(k^{-}|^{\cdot}_{*})$ corresponding to each private key; each actor also knows his own private key.

Figure 7.7(e) defines the personal information known initially about Alice. Each row except the last two shows different context-layer representations of one piece of information, indicating which

Property	Formalisation
Attribute exchange (AX)	$d_1\|^{\zeta}_{u}, d_2?_{gt60}\|^{\zeta}_{u}, d_6\|^{\zeta}_{u}, d_1\|^{\xi}_{u}, d_2?_{gt60}\|^{\xi}_{u}, d_6\|^{\xi}_{u} \in \mathsf{O}_{bs}$
Anonymity revocation (AR)	$*\|^{\kappa}_{al} \leftrightarrow_{\{bs,ii,is,ttp\}} *\|^{\zeta}_{u} \leftrightarrow_{\{bs,ii,is,ttp\}} *\|^{\xi}_{u}$
Irrelevant attribute undetectability (SID)	$d_3\|^{*}_{*} \notin \mathsf{O}_{bs} \wedge d_5\|^{*}_{*} \notin \mathsf{O}_{bs}$
Predicate-attribute undetectability (SPD)	$d_2\|^{*}_{*} \notin \mathsf{O}_{bs}$
IdP attribute undetectability (ID)	$d_1\|^{*}_{*} \notin \mathsf{O}_{is} \wedge d_2\|^{*}_{*} \notin \mathsf{O}_{is} \wedge d_3\|^{*}_{*} \notin \mathsf{O}_{is} \wedge d_2?_{gt60}\|^{*}_{*} \notin \mathsf{O}_{is} \wedge d_5\|^{*}_{*} \notin \mathsf{O}_{ii} \wedge d_6\|^{*}_{*} \notin \mathsf{O}_{ii}$
Mutual IdP involvement undetectability (IM)	$\neg(\exists p : *\|^{\cdot}_{is} \leftrightarrow_{ii} *\|^{p}_{idp2} \wedge *\|^{p}_{u} \leftrightarrow_{ii} *\|^{\kappa}_{al}) \wedge \neg(\exists p : *\|^{\cdot}_{ii} \leftrightarrow_{is} *\|^{p}_{idp1} \wedge *\|^{p}_{u} \leftrightarrow_{is} *\|^{\mu}_{al})$
IdP-SP involvement undetectability (ISM)	$\neg(\exists p : *\|^{\cdot}_{bs} \leftrightarrow_{ii} *\|^{p}_{sp} \wedge *\|^{p}_{u} \leftrightarrow_{is} *\|^{\kappa}_{al}) \wedge \neg(\exists p : *\|^{\cdot}_{bs} \leftrightarrow_{is} *\|^{p}_{sp} \wedge *\|^{p}_{u} \leftrightarrow_{is} *\|^{\mu}_{al})$
Session unlinkability (SL)	$*\|^{\zeta}_{u} \nleftrightarrow_{bs} *\|^{\xi}_{u}$
IdP service access undetectability (IL)	$*\|^{\kappa}_{al} \nleftrightarrow_{ii} *\|^{\zeta}_{u} \wedge *\|^{\kappa}_{al} \nleftrightarrow_{ii} *\|^{\xi}_{u} \wedge *\|^{\mu}_{al} \nleftrightarrow_{is} *\|^{\zeta}_{u} \wedge *\|^{\mu}_{al} \nleftrightarrow_{is} *\|^{\xi}_{u}$
IdP profile unlinkability (IIL)	$*\|^{\kappa}_{al} \nleftrightarrow_{\{ii,is\}} *\|^{\mu}_{al}$
IdP/SP unlinkability (ISL)	$*\|^{\kappa}_{al} \nleftrightarrow_{A} *\|^{\zeta}_{u} \wedge *\|^{\mu}_{al} \nleftrightarrow_{A} *\|^{\zeta}_{u} \wedge *\|^{\kappa}_{al} \nleftrightarrow_{A} *\|^{\xi}_{u} \wedge *\|^{\mu}_{al} \nleftrightarrow_{A} *\|^{\xi}_{u}$ $(A = \{ii, is, bs\})$

Table 7.4: Formalisation of properties in our scenario ($\mathsf{m} \nleftrightarrow_a \mathsf{n}$ means $\neg(\mathsf{m} \leftrightarrow_a \mathsf{n})$; $*$ means for all possible values)

actor initially knows which representation. For instance, d_1 represents a city; Alice knows her city as $d_1|^{\iota}_{al}$ and *ii* knows it as $d_1|^{\kappa}_{al}$. We assume that the actual attribute exchange between user and identity provider during registration has taken place before executing protocol instances π and η, as shown in the κ and μ domains. In the last two rows, each context item represents a different piece of information; e.g., the transaction details $\mathbf{d_7}|^{\zeta}_{u}, \mathbf{d_7}|^{\xi}_{u}$ of the two service provisions are different. We assume some initial knowledge about Alice in the π, η, ζ and ξ domains representing protocols. Knowledge of $i_{ii}|^{\pi}_{u}$, $i_{is}|^{\eta}_{u}$ held by Alice and the respective identity providers represents the fact that Alice has authenticated to them in the context of registration. In the context of the two service provisions, Alice knows that she is the data subject ($i|^{\zeta}_{u}$, $i|^{\xi}_{u}$); the service provider knows transaction details ($\mathbf{d_7}|^{\zeta}_{u}$, $\mathbf{d_7}|^{\xi}_{u}$). Alice knows her own IP address $\mathbf{ip}|^{*}_{u}$, where $* \in \{\pi, \eta, \zeta, \xi\}$; note that it is assumed to change dynamically between sessions.

7.5 Step 2: Model Privacy Properties

Step 2 of our formal privacy comparison is to formalise the properties from Table 7.2 in terms of actor views. Denote the view of an actor $a \in \mathcal{A}$ and a coalition $A \subset \mathcal{A}$ by $V_a = (\mathsf{O}_a, \leftrightarrow_a)$ and $V_A = (\mathsf{O}_A, \leftrightarrow_A)$, respectively. The formalisation of our properties in terms of these views is shown in Table 7.4. AX and AR are detectability and linkability properties (see Section 2.3), respectively.[58] SID, SPD and SID are undetectability properties; SL, IL, IIL, and ISL are unlinkability properties. (Un-)detectability properties are straightforward to formalise; e.g., predicate-attribute undetectability means undetectability by *bs* of the context item $d_2|^{\delta}_{p}$ in any context $*|^{\delta}_{p}$. (Un-)linkability properties translate to contexts not being associable by an actor or coalition. IM and ISM are non-involvement properties: formally, they translate to two associations that should not hold simultaneously; for instance, for IM, there should be no domain p in which *ii* can link the *idp*2 profile to $*|^{\cdot}_{idp2}$ and the u profile to $*|^{\kappa}_{al}$.

[58] For AX, note that *bs* can always associate the personal information of the user to the purchase because of the common context $*|^{\zeta}_{u}$ or $*|^{\xi}_{u}$, so we do not check this.

	Coalition of...				■: undetectable w.r.t. coalition □: detectable w.r.t. coalition							■: involvement unknown			■: unassociable w.r.t. coalition □: associable w.r.t. coalition					
Prop.	*bs*	*ii*	*is*	*ttp*	d_1	d_2	$d_2?_{gt60}$	d_3	d_5	d_6	d_7	*ii*	*is*	*bs*	κ, μ	κ, ζ	κ, ξ	μ, ζ	μ, ξ	ζ, ξ
AX	✓				□		□			□										
SID	✓							■	■											
SPD	✓					■														
ID		✓							■	■										
ID			✓		■	■	■	■												
IM		✓											■							
IM			✓									■								
ISM		✓												■						
ISM			✓											■						
AR	✓	✓	✓	✓												□	□			□
SL	✓																			■
IL		✓														■	■			
IL			✓															■	■	
IIL		✓	✓												■					
ISL	✓	✓	✓													■	■	■	■	

Table 7.5: Schematic overview of the properties in Table 7.4. Each row indicates that with respect to the given coalition of actors, (a) the given items should be (un)detectable; (b) the involvement of the given actors should be unknown; and (c) Alice's profiles in the given domains should be (un)associable

We now analyse whether the above privacy properties cover all privacy risks expressible in our model. To this end, we consider all combinations of coalition and possible knowledge[59]; verify if they represent a privacy risk; and if so, we check by which privacy property they are captured. The result is shown in Table 7.5. The first group of columns indicates the coalition with respect to which a property is defined; the next groups lists the detectability, involvement, and linkability aspects that it entails.

First consider detectability properties. With respect to *bs*, all personal information is required to be either detectable by AX, or undetectable by SID and SPD (except for d_7, which *bs* can always detect by definition of the scenario). Similarly, identity providers can detect attributes they endorse by definition of the scenario, but no others by ID.[60] There are no detectability properties with respect to *ttp*, or about the transaction details d_7. In fact, these aspects would not produce relevant results because *ttp* never learns any attributes, and *bs* never communicates any transaction details.

Involvement properties do not cover *ttp* or *al*: the involvement of *ttp* is publicly known, and Alice's involvement is covered by linkability. For identity providers, there are involvement properties about all remaining parties, i.e., the other identity provider and the service provider. Usually, service providers assess trustworthiness of user attributes by considering which identity provider endorsed them; hence we do not regard involvement properties with respect to the service provider as important.[61]

Linkability properties capture associations by coalitions of actors. Clearly, at least *ii* and *is* are needed to associate κ and μ; IIL states that without help of others, they cannot. There is no property about when *bs* helps them with this; as it turns out, this help never makes a difference. Linkability between user databases and service provisions is defined with respect to the respective identity providers,

[59] In terms of elementary detectability, involvement, and linkability aspects; see Section 2.3

[60] Undetectability of endorsed attributes would be a property of the blind certification feature of Identity Mixer (cf. Bangerter et al. (2004)), as discussed in Section 7.3.

[61] Among the analysed systems, only the Smartcard scheme would satisfy them.

and with respect to a coalition of all identity and service providers. Considering other coalitions would not reveal interesting differences in the systems we analyse. Similarly, no property involves *ii* or *is* in linking the service provisions to each other; in practice, an identity provider would link service provisions to each other by first linking them to its own user profile, which is covered by IL. Finally, AR requires linking the service provisions to κ and not to μ; this is an arbitrary choice made in the definition of the scenario.

7.6 Step 3: Model Communication

Step 3 of our formal privacy comparison is to model the communication in the systems we want to analyse (§7.3). We start by modelling the cryptographic primitives used in the systems as function symbols with construction and elimination rules. For each system, the formalisation then consists of two parts: first, an initial state $\{C_a^0\}_{a \in \mathcal{A}}$ capturing the initial knowledge of all actors (extending Figure 7.7); second, a trace Scenario capturing the communication that takes place in the system during the scenario.

Formal Model of Cryptographic Primitives

To model the systems described in Section 7.3, we extend the model of standard cryptographic primitives from Section 3.4. First, we add zero-knowledge proofs of knowledge, anonymous credentials and their issuing protocols, as discussed in Sections 6.4 and 6.5. Second, we add labelled asymmetric encryption and authenticated key agreement, as described next.

Labelled asymmetric encryption[62] is asymmetric encryption to which a label is unmodifiably attached at encryption time. For instance, the label can represent a policy specifying when the recipient is allowed to decrypt the data. We model labelled asymmetric encryption with function symbol aencl/3, such that $\mathsf{aencl}(\mathsf{m}, \mathsf{l}, \mathsf{pk}(\mathsf{k}^-))$ represents the encryption of message m under public key $\mathsf{pk}(\mathsf{k}^-)$ with label l. The functionality of labelled asymmetric encryption is modelled by the following construction and elimination rules[63]:

[62] Bangerter et al. (2004)

[63] Plus implicit visible failure and reconstruction rules

$$\mathsf{aencl}(m,l,k^+) \leftarrow m,l,k^+ \quad \mathsf{aencl}(m,l,k^+) \rightarrow l \quad \mathsf{aencl}(m,l,pk(k^-)) \xrightarrow{\doteq k^-} m$$

This model is similar to that of normal asymmetric encryption; the label can be derived from the encryption, but not changed. Labelled encryption is a straightforward extension of normal encryption; our model is similar to the one by Camenisch et al.[64].

[64] Camenisch et al. (2010)

In authenticated key agreement (AKA) protocols, two parties derive a unique random session key based on their respective secret keys. We consider the variant presented by Law et al.[65], in which both parties send each other a random value and determine the session key based on this randomness, their own private key, and the other party's public key. We model the session key by function symbol aka/4: $\mathsf{aka}(\mathsf{k}_1^-, \mathsf{n}_1, \mathsf{k}_2^-, \mathsf{n}_2)$ is the session key derived from

[65] Law et al. (2003)

Figure 7.8: Formalisation of smart certificates: initial knowledge and trace

$$\mathcal{C}^0_{al} = (\text{Fig.7.7}) \cup \{\mathsf{sig}(\{i|^{\cdot}_{al}, \mathsf{pk}(k^-|^{\cdot}_{al}), n_c|^{\cdot}_{\cdot}\}, k^-|^{\cdot}_{ca}), k^-|^{\cdot}_{al}, \mathbf{n_{z,a}}|^{\pi}_{\cdot}, \mathbf{n_{z,a}}|^{\eta}_{\cdot}, \mathbf{n_{z,a}}|^{\zeta}_{\cdot}, \mathbf{n_{z,a}}|^{\xi}_{\cdot}\};$$
$$\mathcal{C}^0_{ii} = (\text{Fig.7.7}) \cup \{\mathbf{n_{z,b}}|^{\pi}_{\cdot}, n_a|^{\pi}_{\cdot}\};$$
$$\mathcal{C}^0_{is} = (\text{Fig.7.7}) \cup \{\mathbf{n_{z,b}}|^{\eta}_{\cdot}, n_b|^{\eta}_{\cdot}\};$$
$$\mathcal{C}^0_{bs} = (\text{Fig.7.7}) \cup \{\mathbf{n_{z,b}}|^{\zeta}_{\cdot}, \mathbf{n_{z,b}}|^{\xi}_{\cdot}\}$$

$\mathsf{Scenario} := \mathsf{Reg}_1|^{\pi};\ \mathsf{Reg}_2|^{\eta};\ \mathsf{ServProv}|^{\zeta};\ \mathsf{ServProv}|^{\xi}$

$\mathsf{Reg}_1 :=$

$al(\mathbf{ip}|_u) \to ii(ip|_{idp1})\ :\ \mathsf{sig}(\{i|_u, \mathsf{pk}(k^-|_u), n_c|_{\cdot}\}, k^-|_{ca});$

$al(\mathbf{ip}|_u) \to ii(ip|_{idp1})\ :\ \mathsf{zk}(k^-|_u, \mathsf{pk}(k^-|_u), \mathbf{n_{z,a}}|_{\cdot}, \mathbf{n_{z,b}}|_{\cdot});$

$ii(ip|_{idp1}) \to al(\mathbf{ip}|_u)\ :\ \mathsf{sig}(\{i|_u, d_1|_u, d_2|_u, d_3|_u, n_a|_{\cdot}\}, k^-|_{idp1})$

$\mathsf{Reg}_2 :=$

$al(\mathbf{ip}|_u) \to is(ip|_{idp2})\ :\ \mathsf{sig}(\{i|_u, \mathsf{pk}(k^-|_u), n_c|_{\cdot}\}, k^-|_{ca});$

$al(\mathbf{ip}|_u) \to is(ip|_{idp2})\ :\ \mathsf{zk}(k^-|_u, \mathsf{pk}(k^-|_u), \mathbf{n_{z,a}}|_{\cdot}, \mathbf{n_{z,b}}|_{\cdot});$

$is(ip|_{idp2}) \to al(\mathbf{ip}|_u)\ :\ \mathsf{sig}(\{i|_u, d_5|_u, d_6|_u, n_b|_{\cdot}\}, k^-|_{idp2})$

$\mathsf{ServProv} :=$

$al(\mathbf{ip}|_u) \to bs(ip|_{sp})\ :\ \{\mathsf{sig}(\{i|_u, \mathsf{pk}(k^-|_u), n_c|_{\cdot}\}, k^-|_{ca}), \mathsf{sig}(\{i|_u, d_1|_u, d_2|_u, d_3|_u, n_a|_{\cdot}\}, k^-|_{idp1}), \mathsf{sig}(\{i|_u, d_5|_u, d_6|_u, n_b|_{\cdot}\}, k^-|_{idp2})\};$

$al(\mathbf{ip}|_u) \to bs(ip|_{sp})\ :\ \mathsf{zk}(k^-|_u, \mathsf{pk}(k^-|_u), \mathbf{n_{z,a}}|_{\cdot}, \mathbf{n_{z,b}}|_{\cdot})$

private keys k^-_1, k^-_2 and randomness n_1, n_2. The two construction rules for aka capture how the key can be derived using one private key and the other public key:

$$\mathsf{aka}(k^-_1, n_1, k^-_2, n_2) \leftarrow \mathsf{pk}(k^-_1), n_1, k^-_2, n_2$$
$$\mathsf{aka}(k^-_1, n_1, k^-_2, n_2) \leftarrow k^-_1, n_1, \mathsf{pk}(k^-_2), n_2$$

There are no elimination rules (apart from the implicit ones due to reconstruction).[66] Although the internals of (incorrect) protocols for authenticated key agreement have over the years proven to be a popular target for analysis using formal methods[67], we are not aware of other works that formally model the authenticated key exchange protocols as a primitive.

[66] This model does not take into account that the computation of the session key is actually symmetric, i.e., $\mathsf{aka}(\mathsf{k}^-_1, \mathsf{n}_1, \mathsf{k}^-_2, \mathsf{n}_2)$ and $\mathsf{aka}(\mathsf{k}^-_2, \mathsf{n}_2, \mathsf{k}^-_1, \mathsf{n}_1)$ should be content equivalent. However, as long as no knowledge base contains both messages, this limitation does not cause problems.

[67] E.g., Burrows et al. (1990), Lowe (1996), Paulson (1998)

Smart Certificates

Figure 7.8 displays our formalisation of smart certificates (§7.3). Its first component is the initial state $\{\mathcal{C}^0_x\}_{x \in \mathcal{A}}$. In addition to the knowledge from Figure 7.7, Alice initially knows her public key certificate $\mathsf{sig}(\{i|^{\cdot}_{al}, \mathsf{pk}(k^-|^{\cdot}_{al}), n_c|^{\cdot}_{\cdot}\}, k^-|^{\cdot}_{ca})$ (where $n_c|^{\cdot}_{\cdot}$ represents additional information in the certificate such as the validity date), and the corresponding private key $k^-|^{\cdot}_{al}$. Alice additionally knows contributions $\mathbf{n_{z,*}}|^{*}_{\cdot}$ to her proofs of knowledge of $k^-|^{\cdot}_{al}$; other actors

Figure 7.9: Formalisation of linking service model: initial knowledge and trace

$$\mathcal{C}^0_{ii} = (\text{Fig.7.7}) \cup \{ip|_{ls}, \mathsf{pk}(k^-|_{ls}), i|_{ls},$$
$$i|_{is}, i_{i1,ls}|^{\pi}_{u}, \mathbf{n}|^{\pi}_{\cdot}, i_{ii}|^{\zeta}_{u}, \mathbf{i_{sess}}|^{\zeta}_{u}, \mathbf{n}|^{\zeta}_{\cdot}, i_{ii}|^{\xi}_{u}, \mathbf{i_{sess}}|^{\xi}_{u}, \mathbf{n}|^{\xi}_{\cdot}\};$$
$$\mathcal{C}^0_{is} = (\text{Fig.7.7}) \cup \{ip|_{ls}, \mathsf{pk}(k^-|_{ls}), i|_{ls}, i|_{is}, i_{i2,ls}|^{\eta}_{u}, \mathbf{n}|^{\eta}_{\cdot}\};$$
$$\mathcal{C}^0_{bs} = (\text{Fig.7.7}) \cup \{ip|_{ls}, \mathsf{pk}(k^-|_{ls}), i|_{ls}, i|_{is}\};$$
$$\mathcal{C}^0_{ls} = (\text{Fig.7.7}) \cup \{ip|_{ls}, \mathsf{pk}(k^-|_{ls}), k^-|_{ls}, i|_{ls}, i|_{is}, i_l|^{\nu}_{al}, i_l|^{\pi}_{u}, i_l|^{\eta}_{u}, \mathbf{n'}|^{\zeta}_{\cdot}, \mathbf{n'}|^{\xi}_{\cdot}\}$$

$\mathsf{Scenario} := \mathsf{Reg}_1|^{\pi};\ \mathsf{Reg}_2|^{\eta};\ \mathsf{ServProv}|^{\zeta};\ \mathsf{ServProv}|^{\xi}$

$\mathsf{Reg}_1 :=$

$ii(ip|_{idp1}) \to ls(ip|_{ls})\ :\ \mathsf{sig}(\{i_{i1,ls}|_u, \mathbf{n}|_\cdot\}, \mathbf{k}^-|_{idp})$

$\mathsf{Reg}_2 :=$

$is(ip|_{idp2}) \to ls(ip|_{ls})\ :\ \mathsf{sig}(\{i_{i2,ls}|_u, \mathbf{n}|_\cdot\}, \mathbf{k}^-|_{idp})$

$\mathsf{ServProv} :=$

$ii(ip|_{idp1}) \to bs(ip|_{sp})\ :\ \mathsf{sig}(\{\mathbf{i_{sess}}|_u, d_1|_u, d_2|_u, i|_{ls},$
$\mathsf{aenc}(\{i_{i1,ls}|_u, \mathbf{n}|_\cdot\}, \mathsf{pk}(k^-|_{ls}))\}, k^-|_{idp1});$

$bs(ip|_{sp}) \to ls(ip|_{ls})\ :\ \{\mathsf{aenc}(\{i_{i1,ls}|_u, \mathbf{n}|_\cdot\}, \mathsf{pk}(k^-|_{ls})), \mathsf{sig}(\{\mathbf{i_{sess}}|_u,$
$d_1|_u, d_2|_u, i|_{ls}, \mathsf{aenc}(\{i_{i1,ls}|_u, \mathbf{n}|_\cdot\}, \mathsf{pk}(k^-|_{ls}))\}, k^-|_{idp1})\};$

$ls(ip|_{ls}) \to bs(ip|_{sp})\ :\ \{i|_{idp2}, \mathsf{aenc}(\{i_{i2,ls}|_u, \mathbf{n'}|_\cdot\}, \mathsf{pk}(k^-|_{idp2}))\};$

$bs(ip|_{sp}) \to is(ip|_{idp2})\ :\ \{\mathsf{aenc}(\{i_{i2,ls}|_u, \mathbf{n'}|_\cdot\}, \mathsf{pk}(k^-|_{idp2})), \mathsf{sig}(\{\mathbf{i_{sess}}|_u,$
$d_1|_u, d_2|_u, i|_{ls}, \mathsf{aenc}(\{i_{i1,ls}|_u, \mathbf{n}|_\cdot\}, \mathsf{pk}(k^-|_{ls}))\}, k^-|_{idp1})\};$

$is(ip|_{idp2}) \to bs(ip|_{sp})\ :\ \mathsf{sig}(\{\mathbf{i_{sess}}|_u, d_6|_u\}, k^-|_{idp2})$

similarly know randomness.

The second component of our formalisation of smart certificates is the trace Scenario capturing the communication in our scenario. It consists of traces $\mathsf{Reg}_1|^{\pi}$, $\mathsf{Reg}_2|^{\eta}$, $\mathsf{ServProv}|^{\zeta}$, and $\mathsf{ServProv}|^{\xi}$ corresponding to registration at *ii* and *is*, and two service provisions, respectively. The messages in the traces $\mathsf{Reg}_1|^{\pi}$ and $\mathsf{Reg}_2|^{\eta}$ correspond to those in Figure 7.3(a); the messages in $\mathsf{ServProv}|^{\zeta}$ and $\mathsf{ServProv}|^{\xi}$ correspond to those in Figure 7.3(b). In particular, we model the proofs that Alice knows the secret key corresponding to her public key as ZK proofs with secret information $k^-|_u$ and public information $\mathsf{pk}(k^-|_u)$.

Linking Service Model

Figure 7.9 displays the formalisation of the linking service model (§7.3). This system introduces the linking service *ls* as an additional actor: it has an address and a private/public key pair. *ls* and *is* have publicly known identifiers $i|_{ls}$, $i|_{is}$ used in the referrals. The user database of *ls*, modelled by domain ν, contains an entry for the user containing only the identifier $i_l|^{\nu}_{al}$. User authentication to *ls* during registration is modelled by *ls*'s knowledge of $i_l|^{\pi}_{u}$, $i_l|^{\eta}_{u}$; the pseudonyms generated by the identity providers are modelled as

$\mathcal{C}^0_{al}$ = (Fig.7.7) $\cup$ $\{n_{c1,1}|^{\pi}_{\cdot}, n_{c1,2}|^{\pi}_{\cdot}, n_{c1,3}|^{\pi}_{\cdot}, n_{c1,7}|^{\pi}_{\cdot}, n_{c2,1}|^{\eta}_{\cdot}, n_{c2,2}|^{\eta}_{\cdot}, n_{c2,3}|^{\eta}_{\cdot}, n_{c2,7}|^{\eta}_{\cdot},$
$\mathbf{n_v}|^{\zeta}_{\cdot}, cnd|^{\zeta}_{\cdot}, \mathbf{n}|^{\zeta}_{\cdot}, \mathbf{n_{1,1}}|^{\zeta}_{\cdot}, \mathbf{n_{1,2}}|^{\zeta}_{\cdot}, \mathbf{n_{1,3}}|^{\zeta}_{\cdot}, \mathbf{n_{1,a}}|^{\zeta}_{\cdot}, \mathbf{n_{2,1}}|^{\zeta}_{\cdot}, \mathbf{n_{2,a}}|^{\zeta}_{\cdot}, \mathbf{n_v}|^{\xi}_{\cdot}, cnd|^{\xi}_{\cdot}, \mathbf{n}|^{\xi}_{\cdot}, \mathbf{n_{1,1}}|^{\xi}_{\cdot}, \mathbf{n_{1,2}}|^{\xi}_{\cdot}, \mathbf{n_{1,3}}|^{\xi}_{\cdot}, \mathbf{n_{1,a}}|^{\xi}_{\cdot}, \mathbf{n_{2,1}}|^{\xi}_{\cdot}, \mathbf{n_{2,a}}|^{\xi}_{\cdot}\}$
$\mathcal{C}^0_{ii}$ = (Fig.7.7) $\cup$ $\{n_{c1,4}|^{\pi}_{\cdot}, n_{c1,5}|^{\pi}_{\cdot}, n_{c1,6}|^{\pi}_{\cdot}\}$;
$\mathcal{C}^0_{is}$ = (Fig.7.7) $\cup$ $\{n_{c2,4}|^{\eta}_{\cdot}, n_{c2,5}|^{\eta}_{\cdot}, n_{c2,6}|^{\eta}_{\cdot}\}$;
$\mathcal{C}^0_{bs}$ = (Fig.7.7) $\cup$ $\{\mathbf{n_{1,b}}|^{\zeta}_{\cdot}, \mathbf{n_{2,b}}|^{\zeta}_{\cdot}, \mathbf{n_{1,b}}|^{\xi}_{\cdot}, \mathbf{n_{2,b}}|^{\xi}_{\cdot}\}$

Scenario := $\mathsf{Reg}_1|^{\pi}$; $\mathsf{Reg}_2|^{\eta}$; $\mathsf{ServProv}|^{\zeta}$; $\mathsf{ServProv}|^{\xi}$
Reg_1 :=
$al(\mathbf{ip}|_u) \rightarrow ii(ip|_{idp1})$: $\mathsf{rc}(i|_u, n_{c1,1}|.)$;
$al(\mathbf{ip}|_u) \rightarrow ii(ip|_{idp1})$: $\mathsf{icred}(i|_u, k^-|_{idp1}, \{i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u\}, n_{c1,1}|., n_{c1,2}|., n_{c1,3}|., n_{c1,4}|., n_{c1,5}|., n_{c1,6}|., n_{c1,7}|.)$
Reg_2 :=
$al(\mathbf{ip}|_u) \rightarrow is(ip|_{idp2})$: $\mathsf{rc}(i|_u, n_{c2,1}|.)$;
$al(\mathbf{ip}|_u) \rightarrow is(ip|_{idp2})$: $\mathsf{icred}(i|_u, k^-|_{idp2}, \{d_5|_u, d_6|_u\}, n_{c2,1}|., n_{c2,2}|., n_{c2,3}|., n_{c2,4}|., n_{c2,5}|., n_{c2,6}|., n_{c2,7}|.)$
ServProv :=
$al(\mathbf{ip}|_u) \rightarrow bs(ip|_{sp})$: $\{\mathsf{rc}(i|_u, \mathbf{n}|.), \mathsf{rc}(i_{ii}|_u, \mathbf{n_{1,2}}|.), \mathsf{rc}(d_2|_u, \mathbf{n_{1,1}}|.), \mathsf{rc}(d_3|_u, \mathbf{n_{1,3}}|.),$
$d_1|_u, d_2?_{gt60}|_u, cnd|., \mathsf{pk}(k^-|_{ttp}), \mathsf{aencl}(\{i_{ii}|_u, \mathbf{n_v}|.\}, cnd|., \mathsf{pk}(k^-|_{ttp}))\}$;
$al(\mathbf{ip}|_u) \rightarrow bs(ip|_{sp})$: $\mathsf{zk}(\{\mathsf{cred}(i|_u, k^-|_{idp1}, \{i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u\}, n_{c1,2}|., n_{c1,5}|.), i|_u, i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u, \mathbf{n}|.,$
$\mathbf{n_{1,2}}|., \mathbf{n_{1,1}}|., \mathbf{n_{1,3}}|.\}, \{\mathsf{rc}(\{i|_u, \mathbf{n}|.), \mathsf{rc}(i_{ii}|_u, \mathbf{n_{1,2}}|.), \mathsf{rc}(d_2|_u, \mathbf{n_{1,1}}|.), \mathsf{rc}(d_3|_u, \mathbf{n_{1,3}}|.),$
$d_1|_u, \mathsf{pk}(k^-|_{idp1}), \mathsf{pk}(k^-|_{ttp}), \mathsf{aencl}(\{i_{ii}|_u, \mathbf{n_v}|.\}, cnd|., \mathsf{pk}(k^-|_{ttp})), d_2?_{gt60}|_u\}, \mathbf{n_{1,a}}|., \mathbf{n_{1,b}}|.)$;
$al(\mathbf{ip}|_u) \rightarrow bs(ip|_{sp})$: $\{\mathsf{rc}(i|_u, \mathbf{n}|.), \mathsf{rc}(d_5|_u, \mathbf{n_{2,1}}|.), d_6|_u, cnd|.\}$;
$al(\mathbf{ip}|_u) \rightarrow bs(ip|_{sp})$: $\mathsf{zk}(\{\mathsf{cred}(i|_u, k^-|_{idp2}, \{d_5|_u, d_6|_u\}, n_{c2,2}|., n_{c2,5}|.), i|_u, d_5|_u, d_6|_u, \mathbf{n}|., \mathbf{n_{2,1}}|.\},$
$\{\mathsf{rc}(i|_u, \mathbf{n}|.), \mathsf{rc}(d_5|_u, \mathbf{n_{2,1}}|.), d_6|_u, \mathsf{pk}(k^-|_{idp2})\}, \mathbf{n_{2,a}}|., \mathbf{n_{2,b}}|.)$

Figure 7.10: Formalisation of Identity Mixer: initial knowledge and trace

7

$i_{i1,ls}|^{\pi}_u$ and $i_{i2,ls}|^{\eta}_u$. Alice's authentication at *ii* during service provision is modelled by the fact that *ii* knows the identifiers $i_{ii}|^{*}_u$, $* \in \{\zeta, \xi\}$.

The registration and service provision phases in the trace correspond to Figures 7.4(a) and 7.4(b), respectively. To prove authenticity, the identity providers sign information for *bs* using their private key. *bs* forwards the authentication assertion from *ii* to *ls* and *is* to prove that the user has authenticated. The referrals by *ii* and *is* include random nonces $\mathbf{n}|.$, $\mathbf{n'}|.$ to ensure that *bs* cannot link different sessions by comparing them.

The linking service model aims to satisfy a privacy property specifically about the linking service, which we call *LS attribute undetectability* (LD). We can express this property formally in a similar way to the SID, SPD, and ID properties: $d_1|^{*}_{*} \notin \mathsf{O}_{ls} \wedge ... \wedge d_6|^{*}_{*} \notin \mathsf{O}_{ls}$.

The linking service model in general is independent from message formats. However, the authors also present an instantiation using the SAML 2.0[68] and Liberty ID-WSF 2.0[69] standards. Our model captures that instantiation.

[68] Cantor et al. (2005)
[69] Hodges et al. (2006)

Identity Mixer

The formalisation of the scenario when using Identity Mixer (§7.3) is shown in Figure 7.10. In the trace, registration follows the steps of Figure 7.5(a); service provision is as in Figure 7.5(b). We use Alice's identifier $i|_u^\pi$, which is unknown to the other parties in the system, as the owner identifier in the anonymous credentials. That is, as a result of registration at *ii*, Alice obtains credential

$$\mathsf{cred}(i|_u, k^-|_{idp1}, \{i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u\}, n_{c1,2}|., n_{c1,5}|.),$$

and similarly for registration at *is*. Note that this credential contains Alice's identifier $i_{ii}|_u^\pi$ as an additional attribute: it is used later for anonymity revocation.

As in registration, we use function symbol rc/2 to represent commitments to Alice's secret identifier and attributes. For anonymity revocation purposes, the first message additionally includes a labelled encryption of the identifier $i_{ii}|_u^\pi$ for the trusted third party, with the label $cnd|.$ describing when the anonymity of the transaction may be revoked. The zero-knowledge proof of knowledge in the second message convinces *bs* that:

- Alice owns a credential, signed with *ii*'s private key;
- the owner identifier and attributes in the credential correspond to the values or commitments sent previously;
- the predicate $d_2?_{gt60}|_u$ is satisfied;
- the encrypted message sent previously is encrypted using $\mathsf{pk}(k^-|_{ttp})$ and contains the identifier in the credential.

The second ZK proof is similar. Note that the commitment $\mathsf{rc}(i|_u, \mathbf{n}|.)$ in the first and third messages is the same, guaranteeing to *bs* that the two certificates are indeed of the same user.

Smartcard Scheme

The Smartcard scheme (§7.3) is formalised in Figure 7.11. In this system, the user's personal information is exchanged on her behalf by a tamper-resistant smartcard. The smartcard is modelled as actor *al*. The smartcard has a certified private key; however, this private key is shared between different smartcards so it does not identify the user. Instead, the smartcard has a secret user identifier, modelled by $i|_{al}^{\cdot}$, that is generated on the card and used to generate pseudonyms. The actors *ii*, *is*, and *bs* each have a private key and a corresponding public key certificate signed by the certification authority.

The messages from the registration part of the trace correspond to Figure 7.6(a); the messages from the service provision part correspond to Figure 7.6(b). Parties derive a shared session key using authenticated key agreement based on public key certificates and exchanged randomness. The smartcard generates pseudonyms of Alice with respect to the two identity providers using hashes. In

Figure 7.11: Formalisation of Smartcard scheme: initial knowledge and trace

$$\mathcal{C}^0_{al} = \mathcal{C}^s_{al} \cup \{\mathsf{pk}(k^-|_{ca}), \mathsf{sig}(\{id_c|_\cdot, \mathsf{pk}(k_c^-|_\cdot), n_c|_\cdot\}, k^-|_{ca}), k_c^-|_\cdot, i|_{al}^\cdot,$$
$$\mathbf{n_a}|_\cdot^\pi, \mathbf{n_a}|_\cdot^\eta, \mathbf{n_v}|_\cdot^\zeta, \mathbf{i_{sess}}|_u^\zeta, \mathbf{n_a}|_\cdot^\zeta, \mathbf{n_v}|_\cdot^\xi, \mathbf{i_{sess}}|_u^\xi, \mathbf{n_a}|_\cdot^\xi\};$$
$$\mathcal{C}^0_{ii} = \mathcal{C}^s_{ii} \cup \{\mathsf{pk}(k^-|_{ca}), \mathsf{sig}(\{id|_{ii}, \mathsf{pk}(k^-|_{ii}), n_{ii}|_\cdot\}, k^-|_{ca}), \mathbf{n_b}|_\cdot^\pi\};$$
$$\mathcal{C}^0_{is} = \mathcal{C}^s_{is} \cup \{\mathsf{pk}(k^-|_{ca}), \mathsf{sig}(\{id|_{is}, \mathsf{pk}(k^-|_{is}), n_{is}|_\cdot\}, k^-|_{ca}), \mathbf{n_b}|_\cdot^\eta\};$$
$$\mathcal{C}^0_{bs} = \mathcal{C}^s_{bs} \cup \{\mathsf{pk}(k^-|_{ca}), \mathsf{sig}(\{id|_{bs}, \mathsf{pk}(k^-|_{bs}), n_{bs}|_\cdot\}, k^-|_{ca}),$$
$$\mathbf{n_b}|_\cdot^\zeta, \mathbf{dm}|_\cdot^\zeta, \mathbf{q}|_\cdot^\zeta, \mathbf{n_b}|_\cdot^\xi, \mathbf{dm}|_\cdot^\xi, \mathbf{q}|_\cdot^\xi\}$$

$\mathsf{Scenario} := \mathsf{Reg}_1|^\pi;\ \mathsf{Reg}_2|^\eta;\ \mathsf{ServProv}|^\zeta;\ \mathsf{ServProv}|^\xi$

$\mathsf{Reg}_1 :=$

$al(\mathbf{ip}|_u) \to ii(ip|_{idp1})\ :\ \{\mathsf{sig}(\{id_c|_\cdot, \mathsf{pk}(k_c^-|_\cdot), n_c|_\cdot\}, k^-|_{ca}), \mathbf{n_a}|_\cdot\};$

$ii(ip|_{idp1}) \to al(\mathbf{ip}|_u)\ :\ \{\mathsf{sig}(\{id|_{idp1}, \mathsf{pk}(k^-|_{idp1}), n_{idp1}|_\cdot\}, k^-|_{ca}), \mathbf{n_b}|_\cdot\};$

$al(\mathbf{ip}|_u) \to ii(ip|_{idp1})\ :\ \mathsf{enc}(\mathsf{h}(\{i|_u, id|_{idp1}\}), \mathsf{aka}(k_c^-|_\cdot, \mathbf{n_a}|_\cdot, k^-|_{idp1}, \mathbf{n_b}|_\cdot));$

$ii(ip|_{idp1}) \to al(\mathbf{ip}|_u)\ :\ \mathsf{enc}(\{d_1|_u, d_2|_u, d_3|_u\}, \mathsf{aka}(k_c^-|_\cdot, \mathbf{n_a}|_\cdot, k^-|_{idp1}, \mathbf{n_b}|_\cdot))$

$\mathsf{Reg}_2 :=$

$al(\mathbf{ip}|_u) \to is(ip|_{idp2})\ :\ \{\mathsf{sig}(\{id_c|_\cdot, \mathsf{pk}(k_c^-|_\cdot), n_c|_\cdot\}, k^-|_{ca}), \mathbf{n_a}|_\cdot\};$

$is(ip|_{idp2}) \to al(\mathbf{ip}|_u)\ :\ \{\mathsf{sig}(\{id|_{idp2}, \mathsf{pk}(k^-|_{idp2}), n_{idp2}|_\cdot\}, k^-|_{ca}), \mathbf{n_b}|_\cdot\};$

$al(\mathbf{ip}|_u) \to is(ip|_{idp2})\ :\ \mathsf{enc}(\mathsf{h}(\{i|_u, id|_{idp2}\}), \mathsf{aka}(k_c^-|_\cdot, \mathbf{n_a}|_\cdot, k^-|_{idp2}, \mathbf{n_b}|_\cdot));$

$is(ip|_{idp2}) \to al(\mathbf{ip}|_u)\ :\ \mathsf{enc}(\{d_5|_u, d_6|_u\}, \mathsf{aka}(k_c^-|_\cdot, \mathbf{n_a}|_\cdot, k^-|_{idp2}, \mathbf{n_b}|_\cdot))$

$\mathsf{ServProv} :=$

$al(\mathbf{ip}|_u) \to bs(ip|_{sp})\ :\ \{\mathsf{sig}(\{id_c|_\cdot, \mathsf{pk}(k_c^-|_\cdot), n_c|_\cdot\}, k^-|_{ca}), \mathbf{n_a}|_\cdot\};$

$bs(ip|_{sp}) \to al(\mathbf{ip}|_u)\ :\ \{\mathsf{sig}(\{id|_{sp}, \mathsf{pk}(k^-|_{sp}), n_{sp}|_\cdot\}, k^-|_{ca}), \mathbf{n_b}|_\cdot\};$

$al(\mathbf{ip}|_u) \to bs(ip|_{sp})\ :\ \mathsf{enc}(\mathbf{i_{sess}}|_u, \mathsf{aka}(k_c^-|_\cdot, \mathbf{n_a}|_\cdot, k^-|_{sp}, \mathbf{n_b}|_\cdot));$

$bs(ip|_{sp}) \to al(\mathbf{ip}|_u)\ :\ \{\mathbf{i_{sess}}|_u, \mathsf{enc}(\mathbf{dm}|_\cdot, \mathsf{aka}(k_c^-|_\cdot, \mathbf{n_a}|_\cdot, k^-|_{sp}, \mathbf{n_b}|_\cdot))\};$

$bs(ip|_{sp}) \to al(\mathbf{ip}|_u)\ :\ \{\mathbf{i_{sess}}|_u, \mathsf{enc}(\mathbf{q}|_\cdot, \mathsf{aka}(k_c^-|_\cdot, \mathbf{n_a}|_\cdot, k^-|_{sp}, \mathbf{n_b}|_\cdot))\};$

$al(\mathbf{ip}|_u) \to bs(ip|_{sp})\ :\ \mathsf{enc}(\{d_1|_u, d_2?_{gt60}|_u, d_6|_u, \mathsf{aenc}(\{\mathsf{h}(\{i|_u, id|_{idp1}\}), \mathbf{n_v}|_\cdot\},$
$\mathsf{pk}(k^-|_{ttp}))\}, \mathsf{aka}(k_c^-|_\cdot, \mathbf{n_a}|_\cdot, k^-|_{sp}, \mathbf{n_b}|_\cdot))$

the service provision phase, $\mathbf{q}|_\cdot$ and $\mathbf{dm}|_\cdot$ represent *bs*'s query: what information it needs, and how recent it should be.

Note that in Vossaert et al. (2011), the exact format of the encrypted message to the trusted third party for anonymity revocation is not specified. We chose an encryption of the user's identifier at *ii* because this is most appropriate for our scenario. Also, it is not specified how attributes are sent to the smartcard for caching; we chose to add one additional message to the registration phase containing all attributes.

7.7 Step 4: Verify Privacy Properties

Step 4 of our formal privacy comparison is to verify which properties are satisfied by the analysed systems. This step is performed

Scheme	AX	AR	SID	SPD	ID	IM	ISM	SL	IL	IIL	ISL
Smart certificates	✓	✓	✗	✗	✓	✓	✓	✗	✓	✗	✗
Linking service	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗
Identity Mixer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓†
Smartcard scheme	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 7.6: Comparison of privacy properties claimed and satisfied by the various systems. Filled check-mark: satisfied and claimed; empty check-mark: satisfied and not claimed; filled cross: not satisfied and claimed; empty cross: not satisfied and not claimed (see Table 7.3) . †: may not be satisfiable efficiently depending on non-privacy-related properties.

automatically using our Prolog implementation (Section 3.7): given the formalised properties (§7.5) and communication in the systems (§7.6), it automatically determines which properties hold in which systems.[70] The results are shown in Table 7.6: we now analyse them.

[70] More precisely, it computes the state that the given initial state evolves into by the given trace, also checking trace validity (see Section 6.3); then computes the views of actors and coalitions in this state; and finally, verifies which of the given properties hold in these views.

Non-privacy properties

The two non-privacy properties *attribute exchange (AX)* and *anonymity revocation (AR)* are satisfied in all systems. Indeed, attribute exchange is the basic functional requirement of an IdM system. It is worth noting the relationship between AR and ISL. In smart certificates and the linking service model, ISL does not hold. In this case, AR holds automatically because the service provider and identity providers can link service accesses to user profiles (even without the help of the trusted third party). In the two systems satisfying ISL (the Identity Mixer and Smartcard systems), the transmission of an identifier encrypted for the trusted third party is necessary to satisfy this property.

Detectability properties

The detectability properties with respect to the service provider, *predicate-attribute undetectability (SPD)* and *irrelevant attribute undetectability (SID)*, verify the possibility to reveal predicates about attributes without revealing the exact value; and to reveal some but not all attributes. In smart certificates, the complete certificate is transmitted, so it satisfies neither property. To address SID, the identity provider could issue a separate credential for each user attribute. To partially address SPD, the identity provider could issue several credentials proving common predicates about attributes, e.g. an "age > 60" credential. These latter credentials could be obtained during the service provision phase, in effect transforming smart certificates into a relationship-focused system. Indeed, this variant is discussed in Park and Sandhu (1999). Another possibility is to use certificates that allow efficient proofs of knowledge, as in the Identity Mixer system.

In the linking service model, SPD does not hold. Actually, the linking service model focuses primarily on involvement and linkability issues, leaving the details of the actual attribute exchange to underlying standards. However, in these standards (in particular, SAML) it is not possible to exchange predicates about an attribute instead of its value. Recently, an extension to SAML to achieve this has been proposed[71]. With this extension (or other instantiations),

[71] Neven and Preiss (2011)

the property may hold.

IdP attribute undetectability (ID) and LS attribute undetectability (LD) also do not hold in the linking service model. This is because the linking service and the subscription provider both receive the signed authentication assertion from the address provider as guarantee that the user has logged in. However, in the SAML standard, the attributes are part of this signed message, so they also need to be forwarded. Technically, this could be easily solved by signing the attributes separately from the authentication information. Again, this problem is due to the instantiation of the model with SAML. Note that although ID is not explicitly claimed by the other IdM systems, they do satisfy it.

Involvement properties

The involvement properties state that an identity provider should not know about the user's involvement with other identity providers (*mutual IdP involvement undetectability*, IM) or service providers (*IdP-SP involvement undetectability*, ISM). In credential-focused systems, this is natural: the identity provider issues a credential to the user without involving others, and it is not involved in service provisions. Indeed, smart certificates, Identity Mixer and the Smartcard scheme all satisfy IM and ISM.

In the linking service model, ISM does not hold because there is direct communication between the identity providers and the service provider. In a variant of the model[72], the identity providers and service provider communicate indirectly via the linking service. However, here the identity providers encrypt the attributes for the service provider (to preserve privacy with respect to the linking service), and so still need to know its identity. To prevent this, some kind of trusted intermediary (like the smartcard in the Smartcard scheme) seems to be necessary.

[72] Chadwick and Inman (2009)

Moreover, the linking service model does not satisfy IM. The subscription provider learns from the authentication assertion that the user has an account at the address provider (but not the other way round). This problem is also mentioned in Chadwick and Inman (2009): while "multiple [identity providers] must give [a service provider] the aggregated set of attributes without knowing about one another's involvement", the authors concede that "linked [identity providers] may become aware of just one other [identity provider] – the authenticating [identity provider] – during service provision". IM can be satisfied (within the standards used) if the subscription provider trusts the linking service to verify the address provider's signature. Another possibility to satisfy the property may be to use group signatures[73] for the authentication assertion from the address provider. This solution prevents the subscription provider from learning at which identity provider the user authenticated, but at the cost of reduced accountability.

[73] Chaum and van Heyst (1991)

Linkability properties

Finally, we discuss the results for the linkability properties. *Session unlinkability (SL)* is a natural property for relationship-focused systems, because the identity provider generates a new signature over the attributes at every service provision. Indeed, it holds for the linking service model. It also holds for the credential-focused Identity Mixer system because rather than showing the credential (which would allow linking), the user just proves the validity of properties using ZK proofs. In the Smartcard scheme, the smartcard is trusted to correctly send attributes from the credentials it knows. In the smart certificates scheme, however, the complete credential is shown so the property is not satisfied. *IdP service access unlinkability (IL)*, in contrast, is natural if the identity provider is not involved in service provision, i.e., for the credential-focused smart certificates, Identity Mixer, and Smartcard schemes. It is less natural for relationship-focused systems such as the linking service model. In this case, private information retrieval[74] can be used so that at least the non-authenticating identity provider does not learn which user he is providing attributes of.

[74] Chor et al. (1995)

To achieve *IdP profile unlinkability (IIL)*, global identifiers should be avoided in credential-focused as well as relationship-focused systems. Smart certificates, being based on the user's public key certificate, do not satisfy this property. In Identity Mixer, IIL holds because the identity providers do not learn the identifiers of the credentials they issue. In the Smartcard scheme, it holds because each identity provider learns a different identifier based on a secret known only by the smartcard. In the linking service model, the authenticating identity provider generates a session identifier and includes it in the authentication assertion sent to the other identity provider. This forwarding of the assertion can be avoided if identity providers trust the linking service to verify the authentication assertion: identity providers can then issue attributes under different session identifiers, and the linking service can assert the link between them. However, this only partially solves the problem: identity providers are still both involved in service provision, so they may link using timing information. Indeed, just eliminating global identifiers may not be sufficient to satisfy IIL.

IdP-SP unlinkability (ISL) does not hold for the same two systems that also do not satisfy IIL, and for similar reasons. In smart certificates, all parties learn the user's public key certificate; in the linking service model, the service provider learns the session identifier from the authenticating identity provider. The other systems satisfy it: in Identity Mixer, not even the issuer of the credential can recognise a ZK proof about it; in the Smartcard scheme, the smartcard ensures that the information flow between identity providers and service providers is restricted to just the attributes.

However, as a consequence of ISL holding, extra work is needed to achieve accountability in two respects. First, a message encrypted

to a trusted third party is provided to the service provider to achieve anonymity revocation. Second, although service providers do not learn a credential identifier, they do need assurance that the credential has not been revoked. In the Smartcard scheme, the suggested solution is to let the smartcard perform a regular revocation check. Similarly, in the Identity Mixer system, credentials can be given a short lifetime and be checked for revocation at re-issuing[75]. In both cases, revocation is not immediate.

[75] Camenisch et al. (2009)

For Identity Mixer, two proposals for immediate revocation have been done[76]. The first proposal is to include a serial number in the credential. The credential can be issued so that either the identity provider learns this serial number or not. The former case makes ISL not satisfied. In the latter case, ISL holds but the credential cannot be revoked if the user loses her serial number or does not wish to participate. Depending on the situation at hand, this latter behaviour may not be acceptable. The second proposal is to use a ZK proof that the credential is on a public list of valid credentials[77]. This allows revocation without the user's help while not breaking ISL; however, the user needs to keep track of all revoked credentials in the system, and despite recent advances[78] this may still not be efficient enough. Note that the Smartcard scheme does not support immediate revocation at all.

[76] Camenisch et al. (2006)

[77] Camenisch et al. (2009)

[78] Camenisch et al. (2009)

7.8 Symbolic Analysis of Identity Mixer

In this section, we illustrate how to generalise the above privacy analysis using the symbolic model of Chapter 5. Above, we analyse privacy by determining the knowledge of actors in a specific scenario. In particular, in this scenario, the number of attributes in a credential is fixed to 2 or 3, and all attributes have different contents. Technically, the privacy guarantees derived above only apply in this particular scenario. We now repeat the privacy analysis of one of the above systems, namely Identity Mixer, in our symbolic model. We show how this leads to conclusions that apply beyond a specific scenario.

To perform a symbolic analysis, we first turn the instantiated model of Identity Mixer (Figure 7.10) into a symbolic model. We use the multiple data subjects extension from Section 6.1 for maximal accuracy. For instance, this allows us to model that identifier $i_{ii}|_u^{\pi}$ identifies the user only with respect to the issuing identity provider. The symbolic model, shown in Figure 7.12, consists of three symbolic protocols. *Reg* models a registration protocol in which the issued credential includes the identifier of the credential owner for revocation; *Reg*′ models a registration protocol in which it does not. *Spr* models a service provision protocol. Although *Spr* is independent from the number of attributes in the credentials, it does assume that exactly two credentials are shown, with revocation support for the first one.

We highlight the following aspects of our symbolic formalisa-

$$
\begin{aligned}
Reg(u) &= \{i_{rev}|_{u;idp}, \{d|_{u;\top}\}_{all}, \overline{\mathbf{n_{c,1}}|.}, \mathbf{n_{c,2}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,3}}|.}, \overline{\mathbf{n_{c,7}}|.}, \mathsf{rc}(\mathbf{i}|_{u;\varnothing}, \overline{\mathbf{n_{c,1}}|.}),\\
&\quad \mathsf{icred}(\mathbf{i}|_{u;\varnothing}, \mathbf{k}^{-}|_{idp;\varnothing}, \{i_{rev}|_{u;idp}, \{d|_{u;\top}\}_{all}\}, \overline{\mathbf{n_{c,1}}|.}, \mathbf{n_{c,2}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,3}}|.}, \overline{\mathbf{n_{c,4}}|.}, \mathbf{n_{c,5}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,6}}|.}, \overline{\mathbf{n_{c,7}}|.})\}\\
Reg(idp) &= \{i_{rev}|_{u;idp}, \{d|_{u;\top}\}_{all}, \overline{\mathbf{n_{c,4}}|.}, \mathbf{n_{c,5}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,6}}|.}, \mathsf{rc}(\mathbf{i}|_{u;\varnothing}, \overline{\mathbf{n_{c,1}}|.}),\\
&\quad \mathsf{icred}(\mathbf{i}|_{u;\varnothing}, \mathbf{k}^{-}|_{idp;\varnothing}, \{i_{rev}|_{u;idp}, \{d|_{u;\top}\}_{all}\}, \overline{\mathbf{n_{c,1}}|.}, \mathbf{n_{c,2}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,3}}|.}, \overline{\mathbf{n_{c,4}}|.}, \mathbf{n_{c,5}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,6}}|.}, \overline{\mathbf{n_{c,7}}|.})\}\\
Reg'(u) &= \{i_{rev}|_{u;idp}, \{d|_{u;\top}\}_{all}, \overline{\mathbf{n_{c,1}}|.}, \mathbf{n_{c,2}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,3}}|.}, \overline{\mathbf{n_{c,7}}|.}, \mathsf{rc}(\mathbf{i}|_{u;\varnothing}, \overline{\mathbf{n_{c,1}}|.}),\\
&\quad \mathsf{icred}(\mathbf{i}|_{u;\varnothing}, \mathbf{k}^{-}|_{idp;\varnothing}, \{d|_{u;\top}\}_{all}, \overline{\mathbf{n_{c,1}}|.}, \mathbf{n_{c,2}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,3}}|.}, \overline{\mathbf{n_{c,4}}|.}, \mathbf{n_{c,5}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,6}}|.}, \overline{\mathbf{n_{c,7}}|.})\}\\
Reg'(idp) &= \{\{d|_{u;\top}\}_{all}, i_{rev}|_{u;idp}, \overline{\mathbf{n_{c,4}}|.}, \mathbf{n_{c,5}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,6}}|.}, \mathsf{rc}(\mathbf{i}|_{u;\varnothing}, \overline{\mathbf{n_{c,1}}|.}),\\
&\quad \mathsf{icred}(\mathbf{i}|_{u;\varnothing}, \mathbf{k}^{-}|_{idp;\varnothing}, \{d|_{u;\top}\}_{all}, \overline{\mathbf{n_{c,1}}|.}, \mathbf{n_{c,2}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,3}}|.}, \overline{\mathbf{n_{c,4}}|.}, \mathbf{n_{c,5}}|_{u,idp;\varnothing}, \overline{\mathbf{n_{c,6}}|.}, \overline{\mathbf{n_{c,7}}|.})\}\\
Spr(u) &= \{\overline{\mathbf{n}|.}, cnd|., \{\overline{\mathbf{n_{1,i,p}}|.}\}_{pd1}, \{\overline{\mathbf{n_{1,i,n}}|.}\}_{nd1}, \overline{\mathbf{n_{1,r}}|.}, \overline{\mathbf{n_{1,v}}|.}, \overline{\mathbf{n_{1,a}}|.}, \{\overline{\mathbf{n_{2,i,p}}|.}\}_{pd2}, \{\overline{\mathbf{n_{2,i,n}}|.}\}_{nd2}, \overline{\mathbf{n_{2,a}}|.},\\
&\quad \{\mathsf{rc}(\mathbf{i}|_{u;\varnothing}, \overline{\mathbf{n}|.}), \mathsf{rc}(i_{rev}|_{u;idp1}, \overline{\mathbf{n_{1,r}}|.}), \{\mathsf{rc}(d|_{u;\top}, \overline{\mathbf{n_{1,i,p}}|.})\}_{pd1}, \{\mathsf{rc}(d|_{u;\top}, \overline{\mathbf{n_{1,i,n}}|.})\}_{nd1},\\
&\quad\ \{d|_{u;\top}\}_{d1}, \{d|_{u;\top}\}_{pr1}, cnd|., \mathsf{pk}(\mathbf{k}^{-}|_{ttp;\varnothing}), \mathsf{aencl}(\{i_{rev}|_{u;idp1}, \overline{\mathbf{n_{1,v}}|.}\}, cnd|., \mathsf{pk}(\mathbf{k}^{-}|_{ttp;\varnothing}))\},\\
&\quad \mathsf{zk}(\{\mathsf{cred}(\mathbf{i}|_{u;\varnothing}, \mathbf{k}^{-}|_{idp1;\varnothing}, \{i_{rev}|_{u;idp1}, \{d|_{u;\top}\}_{a1}\}, \mathbf{n_{c,2}}|_{u,idp1;\varnothing}, \mathbf{n_{c,5}}|_{u,idp1;\varnothing}), \mathbf{i}|_{u;\varnothing}, i_{rev}|_{u;idp1},\\
&\quad\ \{d|_{u;\top}\}_{a1}, \overline{\mathbf{n}|.}, \overline{\mathbf{n_{1,r}}|.}, \{\overline{\mathbf{n_{1,i,p}}|.}\}_{pd1}, \{\overline{\mathbf{n_{1,i,n}}|.}\}_{nd1}\}, \{\mathsf{rc}(\mathbf{i}|_{u;\varnothing}, \overline{\mathbf{n}|.}), \mathsf{rc}(i_{rev}|_{u;idp1}, \overline{\mathbf{n_{1,r}}|.}),\\
&\quad\ \{\mathsf{rc}(d|_{u;\top}, \overline{\mathbf{n_{1,i,p}}|.})\}_{pd1}, \{\mathsf{rc}(d|_{u;\top}, \overline{\mathbf{n_{1,i,n}}|.})\}_{nd1}, \{d|_{u;\top}\}_{d1}, \mathsf{pk}(\mathbf{k}^{-}|_{idp1;\varnothing}), \mathsf{pk}(\mathbf{k}^{-}|_{ttp;\varnothing}),\\
&\quad\ maths faencl(\{i_{rev}|_{u;idp1}, \overline{\mathbf{n_{1,v}}|.}\}, cnd|., \mathsf{pk}(\mathbf{k}^{-}|_{ttp;\varnothing})), \{d|_{u;\top}\}_{pr1}\}, \overline{\mathbf{n_{1,a}}|.}, \overline{\mathbf{n_{1,b}}|.}),\\
&\quad \{\mathsf{rc}(\mathbf{i}|_{u;\varnothing}, \overline{\mathbf{n}|.}), \{\mathsf{rc}(d|_{u;\top}, \overline{\mathbf{n_{2,i,p}}|.})\}_{pd2}, \{\mathsf{rc}(d|_{u;\top}, \overline{\mathbf{n_{2,i,n}}|.})\}_{nd2}, \{d|_{u;\top}\}_{d2}, \{d|_{u;\top}\}_{pr2}, cnd|.\},\\
&\quad \mathsf{zk}(\{\mathsf{cred}(\mathbf{i}|_{u;\varnothing}, \mathbf{k}^{-}|_{idp2;\varnothing}, \{d|_{u;\top}\}_{a2}, \mathbf{n_{c,2}}|_{u,idp2;\varnothing}, \mathbf{n_{c,5}}|_{u,idp2;\varnothing}), \mathbf{i}|_{u;\varnothing}, \{d|_{u;\top}\}_{a2},\\
&\quad\ \overline{\mathbf{n}|.}, \{\overline{\mathbf{n_{2,i,p}}|.}\}_{pd2}, \{\overline{\mathbf{n_{2,i,n}}|.}\}_{nd2}\}, \{\mathsf{rc}(\mathbf{i}|_{u;\varnothing}, \overline{\mathbf{n}|.}), \{\mathsf{rc}(d|_{u;\top}, \overline{\mathbf{n_{2,i,p}}|.})\}_{pd2}, \{\mathsf{rc}(d|_{u;\top}, \overline{\mathbf{n_{2,i,n}}|.})\}_{nd2},\\
&\quad\ \{d|_{u;\top}\}_{d2}, \mathsf{pk}(\mathbf{k}^{-}|_{idp2;\varnothing}), \{d|_{u;\top}\}_{pr2}\}, \overline{\mathbf{n_{2,a}}|.}, \overline{\mathbf{n_{2,b}}|.})\}\\
Spr(sp) &= \{\overline{\mathbf{n_{1,b}}|.}, \overline{\mathbf{n_{2,b}}|.}, td|_{u;\top}, \{\mathsf{rc}(\mathbf{i}|_{u;\varnothing}, \overline{\mathbf{n}|.}), \ldots\}, \mathsf{zk}(\ldots), \{\mathsf{rc}(\mathbf{i}|_{u;\varnothing}, \overline{\mathbf{n}|.}), \ldots\}, \mathsf{zk}(\ldots)\}
\end{aligned}
$$

Figure 7.12: Symbolic model of Identity Mixer: registration with and without revocation; service provision with two credentials and revocation

7

tion. First, recall that symbolic protocol roles contain all messages sent, received, and generated by an actor performing that role. For instance, role $Reg(u)$ contains generated nonce $\overline{\mathbf{n_{c,1}}|.}$. Because we assume that authentication has already taken place before registration, we also include $i_{rev}|_{u;idp}$ and $\{d|_{u;\top}\}_{all}$ in $Reg(u)$[79]. Second, nonces $\mathbf{n_{c,2}}|_{u,idp;\varnothing}$, $\mathbf{n_{c,5}}|_{u,idp;\varnothing}$ from registration should not be modelled as instance-random because they end up in the credential that is used in other protocols. Instead, we model them as identifiers of both the user and the identity provider[80]. Identifier $i_{rev}|_{u;idp}$ is modelled as a local identifier with respect to the identity provider: this reflects that different identity providers may internally assign the same identifier to different users. Finally, we use var-lists with different families in the service provision protocol. Namely, we model: all attributes from the first identity provider ($\{d|_{u;\top}\}_{a1}$); disclosed attributes ($\{d|_{u;\top}\}_{d1}$); shown predicates[81] ($\{d|_{u;\top}\}_{pr1}$); attributes of which a predicate is shown ($\{d|_{u;\top}\}_{pd1}$); and non-disclosed attributes ($\{d|_{u;\top}\}_{nd1}$); and their respective nonces. Attributes and nonces from the second identity provider are modelled analogously.

The next step of our symbolic privacy analysis is to construct the constraint graph of the symbolic profiles $Reg(idp)|_u$, $Reg'(idp)|_u$, and $Spr(sp)|_u$. This captures knowledge about a user by arbitrary coali-

[79] We could also leave these messages out, but then they would appear in derivability constraints that are in practice always satisfied, complicating our presentation.

[80] Indeed, these nonces can only occur in protocols relating to both

[81] Note that the symbolic model does not support attribute predicates, so we just model them as data items

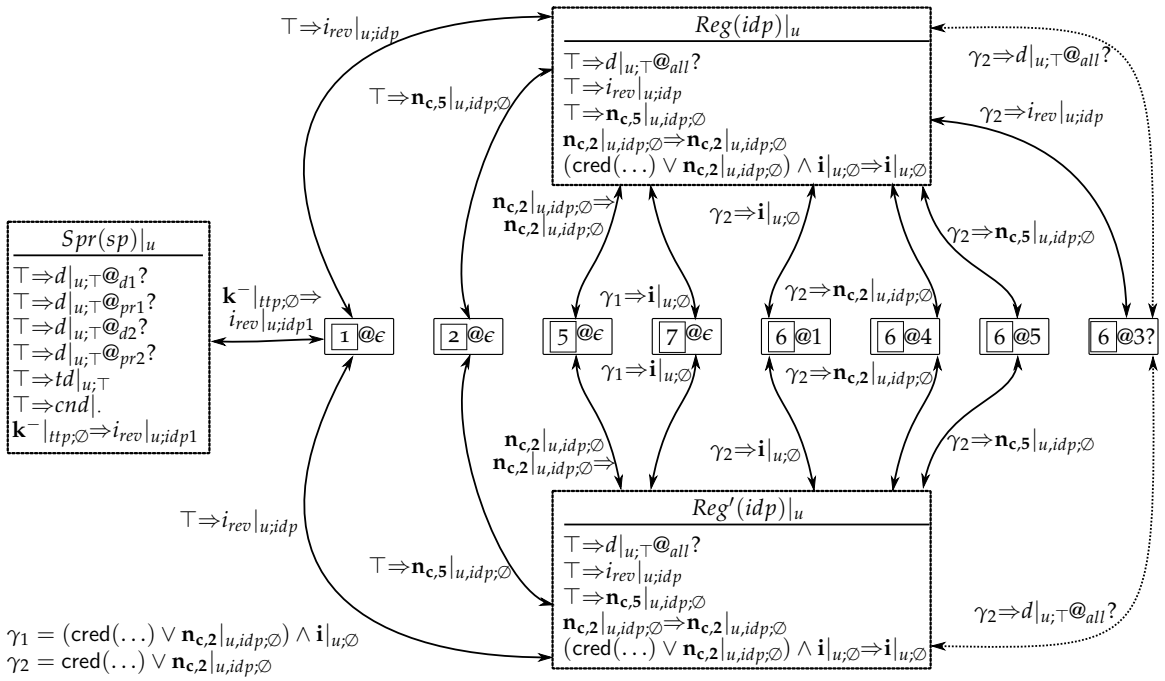


Figure 7.13: Constraint graph for Identity Mixer

tions of identity and service providers that she has interacted with. Figure 7.13 shows the graph, and Figure 7.14 shows the derivation tables on which it is based.[82]

We now briefly discuss what privacy guarantees can be obtained from the constraint graph. First, *irrelevant attribute undetectability* and *predicate-attribute undetectability* hold unconditionally: no satisfiable constraints for $d|_{u;\top}@_F?$, $F \in \{a1, pd1, nd1, a2, pd2, nd2\}$ are listed in profile node $Spr(sp)|_u$, and there are no relevant edges from $Spr(sp)|_u$ to content equivalence nodes. *Session unlinkability* holds as long as the secret key of the TTP is unknown, as reflected by the fact that the only solid outgoing edge from $Spr(sp)|_u$ is labelled with constraint $\mathbf{k}^-|_{ttp;\emptyset}$. For the same reason, also *IdP-SP unlinkability* holds as long as the secrecy key of the TTP is unknown; in any case, the user in the service provision can only be linked to her registration at the first identity provider, since only $i_{rev}|_{u;idp1}$ occurs at an outgoing edge from $Spr(sp)|_u$. *IdP profile unlinkability* can be broken via the internal identifiers $i_{rev}|_{u;idp}$ of the identity providers (node 1 @ϵ); but this link is due to the identity provider's initial knowledge, not due to interaction in Identity Mixer. Other possible links require nonce $\mathbf{n_{c,5}}|_{u,idp;\emptyset}$ to be re-used; or one of the secrets of the user (nonce $\mathbf{n_{c,2}}|_{u,idp;\emptyset}$, the credential, or secret identifier $\mathbf{i}|_{u;\emptyset}$) to be known. Finally, also positive results about knowledge can be obtained. *Attribute exchange* is satisfied because $d|_{u;\top}@_F?$, $F \in \{d1, pr1, d2, p2\}$ have constraint $\mathbf{T}$ in profile node $Spr(sp)|_u$. *Anonymity revocation* is satisfied because $Spr(sp)|_u$ and $Reg(idp)|_u$ can be linked whenever $i_{rev}|_{u;idp1}$ in $Spr(sp)|_u$ coincides with $i_{rev}|_{u;idp}$ in $Reg(idp)|_u$ and the TTP's secret key $\mathbf{k}^-|_{ttp;\emptyset}$ is known. In conclusion, for these properties, we are able to confirm that the privacy guarantees from our instantiated analysis hold in general under reasonable

[82] Note that the credentials from $Reg(idp)|_u$ and $Reg'(idp)|_u$, despite having a different format, may be content equivalent. Namely, this can be the case if the contents of the first attribute in Reg' coincide with those of the identifier in Reg. This would result in symbolic items at different positions in the respective credentials being equatable by the actor, as reflected by the question mark in content equivalence node 6 @3?. This is a consequence of using the var-list extension to our symbolic model; see Section 5.7 for a discussion.

conditions.[83]

7.9 Discussion

This analysis was previously published by Veeningen et al.[84].

The relevance of privacy by data minimisation in the identity management setting is well-established in the literature. It has been recognised as a basic "law of identity" for the design of IdM systems[85]. Hansen et al.[86] argue that privacy-enhancing IdM systems should satisfy a high level of data minimisation with user-controlled linkage of personal data, and by default unlinkability of different user actions. Pfitzmann and Hansen[87] define privacy-enhancing identity management as preserving the unlinkability between user profiles. In the recent PrimeLife project[88], the use of credentials to control identities has been studied. Finally, in a general survey, Alpár et al.[89] identify three main privacy issues in identity management: linkability across domains, identity providers knowing user transactions, and violation of proportionality and subsidiarity (i.e., the exchange of minimal information needed for a certain goal). These three issues correspond to our three kinds of privacy properties: linkability, involvement, and detectability, respectively. In contrast to the vision of minimising actor knowledge, Landau and Moore[90] argue that preventing service providers from collecting transaction data may not be desirable because it prevents the adoption of IdM systems in practice . This falls into a broader discussion on incentives of participants in IdM systems[91] that is out of scope for this work.

The formal analysis presented in this chapter aims to improve the way privacy by data minimisation is assessed compared to existing comparisons such as those by the Independent Centre for Privacy Protection Schleswig-Holstein[92] and Hoepman et al.[93]. Both these comparisons consider data minimisation as one aspect of a much more general comparison of IdM systems. Data minimisation properties are specified in a high-level way, and verified manually by inspecting the user interface and documentation of the systems. The study by the Independent Centre for Privacy Protection Schleswig-Holstein considers three different criteria: "usage of pseudonyms/anonymity"; "usage of different pseudonyms" and "user [is] only asked for needed data" (judged on a yes/no scale). Hoepman et al. consider two: "directed identity"/"pseudonymous/anonymous use" and "minimal disclosure" (judged on a ++ to -- scale). To improve the objectivity and accuracy of such assessments, scores for such criteria may instead be obtained by aggregating formal analysis results like ours. To obtain a better understanding of privacy differences, these formal results can then be analysed as in Section 7.7. Note that we only assess privacy given what information should be exchanged; to verify if this exchange of information is really needed, or consented to by the user, other methods[94] should be used. Some other aspects of the privacy as-

[83] Note that, due to slight technical differences, the instantiated model of Identity Mixer from Section 7.6 is not an instantiation of the present symbolic model in the sense of Definition 5.1.3. In particular, the instantiated model uses the attribute predicate extension of the instantiated model which we have not defined in the symbolic setting. Hence, the privacy guarantees in the instantiated model technically do not follow from this symbolic analysis. We leave the more "purist" approach in which the models directly correspond as future work.

[84] Veeningen et al. (2014)

[85] Cameron (2006)

[86] Hansen et al. (2004)

[87] Pfitzmann and Hansen (2009)

[88] See, e.g., Ardagna et al. (2010a,b)

[89] Alpár et al. (2011)

[90] Landau and Moore (2011)

[91] E.g., Anderson (2011), Camp (2010)

[92] Independent Centre for Privacy Protection Schleswig-Holstein (2003)

[93] Hoepman et al. (2008)

[94] E.g., Compagna et al. (2009)

sessments in Independent Centre for Privacy Protection Schleswig-Holstein (2003), Hoepman et al. (2008) seem less suitable for formal verification, e.g. the user-friendliness and the use of standards in the systems.

Some more formal works on privacy in identity management are available. Pfitzmann and Hansen[95] define privacy-enhancing identity management as preserving unlinkability between different user profiles, and explore the meaning of linkability and its relationship with related concepts in a semi-formal way. Their informal definitions formed the inspiration for the model presented in Chapter 2. Other formal work on identity management has mainly focused on safety properties with respect to misbehaving attackers, rather than privacy properties with respect to insiders who follow the protocol specification. In this context, Li et al.[96] and Suriadi[97] consider unlinkability properties for Identity Mixer and related anonymous credential schemes; Camenisch et al.[98] consider undetectability properties. For SAML[99], a standard for the exchange of identity information between identity and service providers used in the linking service model, Armando et al.[100] consider secrecy properties. Our work differs from this latter category in two respects: first, we define properties in a general setting, allowing comparisons between different systems; and second, we distinguish between the roles of different insiders rather than considering one outsider, enabling us to express which (coalitions of) actors can associate or detect certain information.

In this work, we focus on minimising knowledge of personal information by technical means; other works address other aspects of privacy in identity management. Landau et al.[101] argue that privacy protection can be achieved not just technically, but also by legal and policy means[102]. Hansen et al.[103] argue that apart from ensuring data minimisation, privacy-enhancing IdM systems should also make the user aware of what information is exchanged about her and who can link it; and allow the user to control these aspects. Bhargav-Spantzel et al.[104] stress the importance of trust between different parties in identity management, and in particular, trust of the user in other parties' handling of her personal information. Our method can complement this demand for transparency by providing a precise view on how the choice for an IdM system impacts privacy.[105]

[95] Pfitzmann and Hansen (2009)

[96] Li et al. (2009)

[97] Suriadi (2010)

[98] Camenisch et al. (2010)

[99] Cantor et al. (2005)

[100] Armando et al. (2008)

[101] Landau et al. (2009)

[102] For instance, see di Vimercati et al. (2011) on specification and enforcement of policies

[103] Hansen et al. (2004)

[104] Bhargav-Spantzel et al. (2007b)

[105] Interestingly, recent research in behavioural economics suggests that offering transparency to users might actually reduce their privacy by inducing them to release more information, e.g., see Brandimarte et al. (2010).

Constraint	Message	CE
$\top$	$d\|_{u;\top}@_{all}?$	1
$\top$	$i_{rev}\|_{u;idp}$	1
$\top$	$\mathbf{n_{c,5}}\|_{u,idp;\varnothing}$	2
$\top$	$\mathsf{pk}(\mathbf{k}^-\|_{idp;\varnothing})$	3
$\top$	$\mathbf{k}^-\|_{idp;\varnothing}$	4
$\mathbf{n_{c,2}}\|_{u,idp;\varnothing}$	$\mathbf{n_{c,2}}\|_{u,idp;\varnothing}$	5
$\mathsf{cred}(\ldots) \vee \mathbf{n_{c,2}}\|_{u,idp;\varnothing}$	$\mathsf{cred}(\mathbf{i}\|_{u;\varnothing}, \mathbf{k}^-\|_{idp;\varnothing}, \{i_{rev}\|_{u;idp}, \{d\|_{u;\top}\}_{all}\}, \mathbf{n_{c,2}}\|_{u,idp;\varnothing}, \mathbf{n_{c,5}}\|_{u,idp;\varnothing})$	6
$(\mathsf{cred}(\ldots) \vee \mathbf{n_{c,2}}\|_{u,idp;\varnothing}) \wedge \mathbf{i}\|_{u;\varnothing}$	$\mathbf{i}\|_{u;\varnothing}$	7

(a) Derivation table for $Reg(idp)$

Constraint	Message	CE
$\top$	$d\|_{u;\top}@_{all}?$	1
$\top$	$i_{rev}\|_{u;idp}$	1
$\top$	$\mathbf{n_{c,5}}\|_{u,idp;\varnothing}$	2
$\top$	$\mathsf{pk}(\mathbf{k}^-\|_{idp;\varnothing})$	3
$\top$	$\mathbf{k}^-\|_{idp;\varnothing}$	4
$\mathbf{n_{c,2}}\|_{u,idp;\varnothing}$	$\mathbf{n_{c,2}}\|_{u,idp;\varnothing}$	5
$\mathsf{cred}(\ldots) \vee \mathbf{n_{c,2}}\|_{u,idp;\varnothing}$	$\mathsf{cred}(\mathbf{i}\|_{u;\varnothing}, \mathbf{k}^-\|_{idp;\varnothing}, \{d\|_{u;\top}\}_{all}, \mathbf{n_{c,2}}\|_{u,idp;\varnothing}, \mathbf{n_{c,5}}\|_{u,idp;\varnothing})$	6
$(\mathsf{cred}(\ldots) \vee \mathbf{n_{c,2}}\|_{u,idp;\varnothing}) \wedge \mathbf{i}\|_{u;\varnothing}$	$\mathbf{i}\|_{u;\varnothing}$	7

(b) Derivation table for $Reg'(idp)$

Constraint	Message	CE
$\top$	$d\|_{u;\top}@_{d1}?$	1
$\top$	$d\|_{u;\top}@_{pr1}?$	1
$\top$	$d\|_{u;\top}@_{d2}?$	1
$\top$	$d\|_{u;\top}@_{pr2}?$	1
$\top$	$td\|_{u;\top}$	1
$\top$	$cnd\|_{\cdot}$	1
$\top$	$\mathsf{pk}(\mathbf{k}^-\|_{ttp;\varnothing})$	3
$\top$	$\mathsf{pk}(\mathbf{k}^-\|_{idp1;\varnothing})$	3
$\top$	$\mathsf{pk}(\mathbf{k}^-\|_{idp2;\varnothing})$	3
$\mathbf{k}^-\|_{ttp;\varnothing}$	$i_{rev}\|_{u;idp1}$	1
$\mathbf{k}^-\|_{ttp;\varnothing}$	$\mathbf{k}^-\|_{ttp;\varnothing}$	4
$\mathbf{k}^-\|_{idp1;\varnothing}$	$\mathbf{k}^-\|_{idp1;\varnothing}$	4
$\mathbf{k}^-\|_{idp2;\varnothing}$	$\mathbf{k}^-\|_{idp2;\varnothing}$	4

(c) Derivation table for $Spr(sp)$

Figure 7.14: Derivation tables for the symbolic model of Identity Mixer

8
Assessing Data Minimisation of Patient Pseudonyms

Contents

THE SECOND DEMONSTRATION of our privacy analysis framework is a study on the privacy consequences of pseudonymising medical data for research purposes. The quality of medical research benefits from the collection of patient data from different health care organisations. By analysing data from different sources, researchers are able to study treatments from several angles, which can lead to new insights. To facilitate the collection and dissemination of medical data, initiatives like the Dutch Parelsnoer initiative have developed data management infrastructures[1]. Such infrastructures store patient data collected from health care organisations into a central medical research database and then distribute such data to researchers. Besides providing data to researchers, they should also allow findings by researchers to be returned to hospitals to facilitate treatment.

[1] E.g., Parelsnoer Initiatief (2008, 2009), Pommerening and Reng (2004)

When distributing patient data, these infrastructures should protect the patient's privacy by making sure that data are properly anonymised. In particular, researchers should not be able to link data to a particular patient, or data from different research projects to each other. However, it is not possible to just remove all identifiers from the data: the need to share findings with the patient's hospital implies that the data may need to be deanonymised. Deanonymisation should only be possible following a rigorous process involving multiple parties; the infrastructure should technically ensure that, apart from this process, there is no way to correlate patient data.

In this chapter, we demonstrate that the impact on privacy of using such infrastructures can be analysed using our framework.

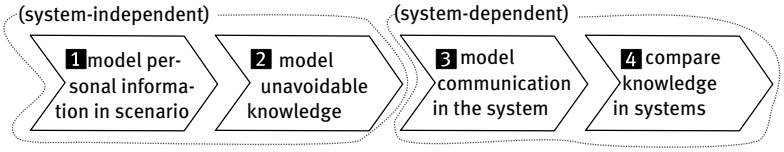


Figure 8.1: Steps of a data minimisation analysis using our framework

However, we use a fundamentally different approach than in the previous chapter. Namely, in the previous chapter, we analysed privacy by formalising privacy properties capturing what information should *not* be learned by the actors in the system. In this chapter, we formalise what information *should* be learned by the actors to implement the functionality of the system, and then use coalition graphs (Section 2.4) to find additional, hence avoidable, knowledge. In other words, we assess systems' satisfaction of the "data minimisation"[2] principle; namely, that actors only learn the information that they need to perform their tasks in the system.

[2] OECD (2002)

The steps needed to perform such a data minimisation analysis are shown in Figure 8.1. These steps are similar to those for verifying privacy properties (cf. Figure 7.1). As before, the *first step* is to model the relevant personal information. Now, the *second step* is to model which personal information needs to be known by the actors in the system to perform their tasks. Unlike in the previous chapter, these two steps are performed in terms of information rather than context-layer representations, and hence simpler. They are system-independent; however, what information is considered "needed" by actors does depend on general assumptions about what kind of systems are considered.

The *third step*, namely, modelling the exchange of information in each system considered, is as before. The *fourth step* again is different: instead of returning answers about a fixed set of privacy properties, our privacy analysis tool (Section 3.7)[3] now returns coalition graphs. Namely, it returns coalition graphs summarising the knowledge of actors in each system; and combined coalition graphs comparing different systems to each other and to the "needed" information identified in step two. Hence, we find potential privacy drawbacks of systems, which may lead to suggestions for improvements.

[3] The tool and the formal models of systems presented in this chapter are available at `http://code.google.com/p/objective-privacy/`

8

Outline In this chapter:

- We discuss privacy issues in the pseudonymisation of patient data for research, and define the scope of our analysis (§8.1);
- We perform a data minimisation analysis using the steps described above, and analyse the results (§8.2–§8.5);
- We discuss and model possible improvements suggested by the analysis (§8.6);
- We finish by discussing relevant (de)pseudonymisation proposals and analyses (§8.7).

8.1 Pseudonymisation Infrastructures

In this section, we analyse the setting for our comparison, and introduce the systems we analyse.

We first identify *functional requirements* **(FR*)** stating what personal information needs to be known in any system for pseudonymising patient information. We assume that the system complies with the Dutch legal framework, which requires health care organisations to use the "burgerservicenummer" (BSN; the Dutch Social Security Number) to store medical data for treatment **(FR1)**. As a result of using the system, the researchers should learn the patient information **(FR2)**. In certain circumstances, it should also be possible to link data from the researcher to the patient. Namely, in case of a discovery beneficial for the patient (a so-called *coincidental finding*), the health care organisations which collected the data should be notified so they can provide treatment: *full depseudonymisation* **(FR3)**. Moreover, if additional patient data is needed for a certain research project, it should be possible to link the additional data about a patient to data from a previously distributed dataset: *partial depseudonymisation* **(FR4)**.

We narrow the scope of our analysis by assuming several *design decisions* **(DD*)**. To facilitate the provision of medical data to researchers, data collected from different health care organisations can be stored into a single database[4], hereafter called *Central Infrastructure* (CI). We assume that there is such a CI, which stores data about one patient from different hospitals in one record **(DD1)**. It obtains this data directly from the different hospitals **(DD2)**. When a researcher needs a dataset, the CI compiles it from its database and sends it to the researcher **(DD3)**, who is not otherwise involved in the pseudonymisation process. For extension of a dataset (i.e., partial depseudonymisation), the researcher contacts the CI, which then compiles the extended dataset without involving the original hospital **(DD4)**. Finally, depseudonymisation should be performed via a trusted third party to ensure that it is only possible under strictly defined conditions **(DD5)**.

[4] See, e.g., Parelsnoer Initiatief (2008, 2009), Pommerening and Reng (2004)

In this chapter, we analyse two particular (de)pseudonymisation infrastructures developed by the Parelsnoer initiative[5], a collaboration between eight university medical centres in the Netherlands. One infrastructure is based on hashing, and one is based on a trusted "pseudonymisation service". We now discuss both.

[5] Parelsnoer Initiatief (2008, 2009)

Hash-Based Pseudonymisation Infrastructure (H-PI) The first Parelsnoer proposal for (de)pseudonymisation[6] uses pseudonyms for the storage and transmission of medical data that are constructed using hash functions (Figure 8.2). In particular, when providing data to the CI, hospitals use a hash h_1 of a patient's BSN and birth date as pseudonym. This allows the CI to link data from different hospitals without learning the BSN. Each research project has a separate identifier; when the CI distributes data for a research project, the

[6] Parelsnoer Initiatief (2008)

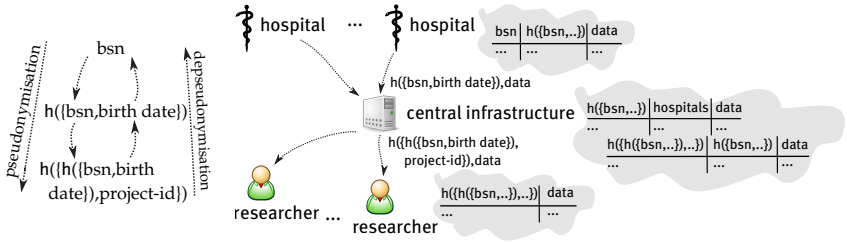


Figure 8.2: Parelsnoer Hash-Based Pseudonymisation Infrastructure (H-PI): pseudonyms (left) and operation (right)

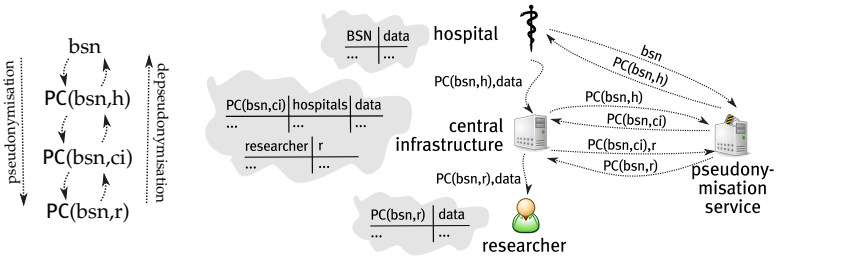


Figure 8.3: Parelsnoer Infrastructure with Pseudonymisation Service (PS-PI): pseudonyms (left) and operation (right)

project identifier is hashed along with the pseudonym h_1 into a new pseudonym h_2. For partial depseudonymisation, the CI needs a table containing the links (h_1, h_2) for all distributed datasets. For full depseudonymisation, the CI additionally needs a table containing the identities of hospitals for all patient pseudonyms h_1. Each hospital stores a table containing the links (bsn, h_1) for its own patients.

One drawback of this approach is that an attacker who learns a pseudonym, can try to depseudonymise it using a dictionary attack: this is feasible because the entropy in the combination of BSN and birth date is at most 42 bits[7]. In addition, the fact that hospitals and CI need to keep pseudonym translation tables poses significant risks of data breaches. Note that H-PI does not use a TTP to control depseudonymisation; as shown later, this makes it non-optimal in terms of data minimisation.

[7] Parelsnoer Initiatief (2008)

Pseudonymisation Service Infrastructure (PS-PI) The Pseudonymisation Service Infrastructure[8] of Parelsnoer addresses the limitations of the hash-based approach using a TTP called *pseudonymisation service* (PS). The pseudonyms used in the system are called "pseudocodes". These pseudocodes are unique given a BSN and a "domain" (i.e., the CI, hospitals, and research projects) in which patient data should be linked. The mapping between BSNs and pseudocodes and between pseudocodes from different domains is calculated using a secret known only by the PS.

[8] Parelsnoer Initiatief (2009)

Figure 8.3 shows the translation steps (left) and the information that is exchanged and stored (right). First, the PS translates the BSN into a pseudocode in the hospital domain, which the hospital uses to send medical data to the CI. The CI requests the PS to re-translate the pseudocode to its own domain so it can link data from different hospitals together. When data are distributed to a researcher, the pseudocode is translated to the project domain. For depseudonymisation, pseudocodes are translated back to the BSN in exactly the opposite order. For partial depseudonymisation, the researcher provides the research pseudocode to the CI, which requests the PS to translate it to its own domain. The CI remembers which research do-

main belongs to which researcher; it includes the research domain in the depseudonymisation request to the PS, which compares it to the actual domain in the pseudocode. For full depseudonymisation, the CI asks the PS to translate this pseudocode from the CI domain to the hospital domain based on the list of hospitals that have provided data.

This infrastructure solves the drawbacks of the hash-based infrastructure. Since pseudocodes are calculated using a secret known only by the PS, this infrastructure is not subject to dictionary attacks. Moreover, the hospital and CI no longer store tables to translate pseudocodes to BSNs. Indeed, depseudonymisation is not possible without the PS, reducing privacy impact when data of hospitals and the CI are compromised.

8.2 Step 1: Model Personal Information

We now analyse data minimisation in the above two systems by following the steps outlined at the beginning of this chapter. The first step is to model the personal information in the system. Because coalition graphs are defined at the information layer of our three-layer model of personal information (see Chapter 2), it suffices to give an information-layer representation. We consider a scenario that is general enough to capture all aspects we are interested in, yet small enough to allow a clear visualisation.

We consider six different actors: three hospitals umc_1, umc_2, and umc_3; one researcher r; and CI ci and TTP ttp (in PS-PI: the PS). These actors exchange information about a particular patient. Two of the three hospitals hold medical data about the patient: umc_1 knows three pieces of information d_1, d_2, and d_3; umc_2 knows d_4, d_5, and d_6. The items d_i are non-identifying; i.e., they represent attributes for which different patients may have a common value. The hospitals identify their patient records by BSN bsn. The third hospital umc_3 does not hold information about the patient. Researcher r needs data about the patient for two different research projects: d_1 and d_4 for one project, and d_2 and d_5 for a second project. By considering two hospitals with patient data and one without, we can consider correlation between these two types of hospitals and other actors, and between two different hospitals that both know the patient.[9]

Our scenario has three steps. First, umc_1 and umc_2 provide their patient data to ci. Second, r receives patient data from the ci in two different datasets for the two different projects. In both steps, the TTP may be involved. Third, as part of the investigation in the first research project, the researcher learns a coincidental finding d_7 that may be important for treatment of the patient. We consider the moment when the coincidental finding has been made, but depseudonymisation has not been performed yet. In particular, the hospitals do not know d_7 yet, so we can reason about coalitions that enable hospitals to link d_7 to the corresponding patient.

[9] Verifying privacy with respect to a single researcher involved in two different projects is sufficient: if one researcher in two projects cannot link the data, then neither can two researchers in two projects.

Requirement/Design decision	Privacy consequences
(FR1) Hospitals store data using BSN	$\{umc_1\} \vDash \{bsn, d_1, d_2, d_3\}$, $\{umc_2\} \vDash \{bsn, d_4, d_5, d_6\}$
(FR2) Researchers obtain dataset	$\{r\} \vDash \{d_1, d_4, d_7\}$, $\{r\} \vDash \{d_2, d_5\}$
(FR3) Full depseudonymisation	$\{umc_1, ci, ttp, r\} \vDash \{bsn, d_7\}$, $\{umc_2, ci, ttp, r\} \vDash \{bsn, d_7\}$
(FR4) Partial depseudonymisation	$\{umc_1, ci, ttp, r\} \vDash \{d_1, d_2, d_3, d_7\}$, $\{umc_2, ci, ttp, r\} \vDash \{d_4, d_5, d_6, d_7\}$
(DD1) CI collects data	$\{ci\} \vDash \{d_1, d_2, d_3, d_4, d_5, d_6\}$, $\{umc_1, ci, ttp\} \vDash \{bsn, d_1, d_2, d_3, d_4, d_5, d_6\}$, $\{umc_2, ci, ttp\} \vDash \{bsn, d_1, d_2, d_3, d_4, d_5, d_6\}$
(DD2) Data transfer between hospital, CI	$\{{umc_1}^*, ci^*\} \vDash \{bsn, d_1, d_2, d_3, d_4, d_5, d_6\}$, $\{{umc_2}^*, ci^*\} \vDash \{bsn, d_1, d_2, d_3, d_4, d_5, d_6\}$
(DD3) Dataset from CI to researcher	$\{ci^*, r\} \vDash \{d_1, d_2, d_3, d_4, d_5, d_6, d_7\}$
(DD4) Partial depseudo without hospital	$\{ci, ttp, r\} \vDash \{d_1, d_2, d_3, d_4, d_5, d_6, d_7\}$
(DD5) (De)pseudonymisation by TTP	(See consequences of **(FR)**, **(DD)**)

Table 8.7: Privacy consequences of functional requirements **(FR)** and design decisions **(DD)**

8.3 Step 2: Model Unavoidable Knowledge

The second step of analysing data minimisation using coalition graphs, is to analyse the optimal privacy achievable in the given setting. Namely, we define an "optimal" coalition graph formalising the unavoidable privacy consequences of functional requirements (expressing information exchange needed for the functionality of the system) and design decisions (expressing general assumptions about the type of system we consider).

When modelling privacy consequences, we distinguish between *honest* actors and *curious* actors. *Honest* actors store only the information the system allows them to store; *curious* remember all information they have observed. Which privacy consequences are unavoidable depends on which actors we assume to be curious. For instance, the PS-PI architecture aims to ensure that depseudonymisation can only happen though the PS. However, this can only be ensured when the other actors are honest: if hospitals and the CI are curious, they can link data by remembering pseudocodes, bypassing the PS. We denote honest actors by $\{umc_1, umc_2, \ldots\}$, and curious actors by $\{{umc_1}^*, {umc_2}^*, \ldots\}$, and consider coalitions in which each actor is either honest or curious. We do not distinguish between honest and curious researchers: they are external parties for which it is hard to enforce honest behaviour.

We formalise privacy consequences in terms of record detectability $A \vDash O$ of records $O \subset \{bsn, d_1, d_2, d_3, d_4, d_5, d_6, d_7\}$ with respect to coalitions A of honest or curious actors from $\{umc_1, umc_2, umc_3, ci, ttp, r\}$. Actors' knowledge is taken after the CI has distributed the datasets to the researcher and she has made a coincidental finding, but before depseudonymisation has taken place. Table 8.7 shows our formalisations.

Functional requirements **(FR1)** and **(FR2)** directly translate to the fact that hospitals and researchers know certain data about the patient. Functional requirements **(FR3)** and **(FR4)** state that full/partial depseudonymisation should be possible. In particular, a hospital, the TTP, the CI, and the researcher together should be able to perform full depseudonymisation, i.e., they should be able to link d_7 to bsn. Similarly, for partial depseudonymisation, they should be able

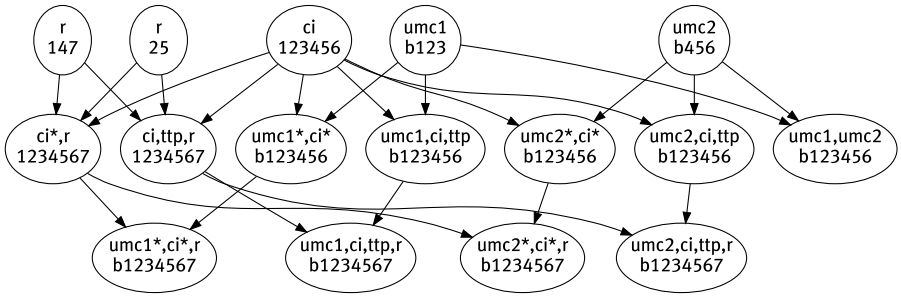


Figure 8.4: Coalition graph of optimal situation (O-PI). Node captions represent coalitions A and records O, respectively, with $A \vDash O$; 'b' means bsn, '1' means d_1, etc.

to link d_7 to the patient data.

Introducing the medical research database CI has several privacy consequences. Design decision **(DD1)** states that the task of the CI is to collect and link the data from different hospitals; it has two consequences. First, the CI knows the medical data from the two hospitals in one record. Second, if a hospital, CI and TTP combine their knowledge, they can link the BSN to the full patient record at the CI (by definition of the collection process). By design decision **(DD2)**, we consider systems where the CI and hospital communicate directly during the collection process. At the time of this communication, the hospital knows the BSN, and the CI knows the link to the full patient record. Therefore, if both refer back to this communication, they can link the BSN to the full patient record without the PS. Design decision **(DD3)** states that the researcher is involved in (de)pseudonymisation merely as the passive recipient of the datasets. During the provision of such a dataset, the CI knows the link between records in the distributed dataset and the full patient records. If the CI is curious and remembers this link, and the researcher discovers an accidental finding related to some record, then together they can link the finding to the record. Design decision **(DD4)** states that hospitals are not involved in partial depseudonymisation; instead, it is performed by linking the incidental finding of the researcher to the patient record at the CI using the TTP. Finally, design decision **(DD5)** is the introduction of the TTP. This design decision is reflected by the fact that TTP is needed for data collection **(DD1)** and full **(FR3)** and partial **(FR4)**, **(DD4)** depseudonymisation.

Figure 8.4 combines the privacy consequences of Table 8.7 into a coalition graph. Intuitively, it is the coalition graph of a hypothetical infrastructure O-PI which satisfies all requirements and design decisions, and whose design is optimal in terms of data minimisation. Nodes represent unavoidable disclosures. For clarity, we identify honest and non-honest actors for the "subcoalition" relation. For instance, umc_2 is a subcoalition of $\{umc_2^*\, ci^*\}$, so we have an edge from $\{umc_2\} \vDash \{bsn, d_4, d_5, d_6\}$ to $\{umc_2^*, ci^*\} \vDash \{bsn, d_1, d_2, d_3, d_4, d_5, d_6\}$.

To obtain the graph, we considered which consequences from Table 8.7 apply to any particular coalition. Given a coalition A, we consider which record detectability statements $A \vDash O$ follow from

the entries in the table. For instance, for coalition $A = \{umc_1\}$, the table implies detectability of record $\{bsn, d_1, d_2, d_3\}$, which corresponds to a node in the graph. Coalition $A = \{r\}$ can detect two records $\{d_1, d_4, d_7\}$ and $\{d_2, d_5\}$ but it should not be able to link them together, so the two records occur as two nodes in the graph. On the other hand, for coalition $A = \{umc_1, umc_2\}$, $A \vDash \{bsn, d_1, d_2, d_3\}$ follows from $\{umc_1\} \vDash \{bsn, d_1, d_2, d_3\}$, and $A \vDash \{bsn, d_4, d_5, d_6\}$ follows from $\{umc_2\} \vDash \{bsn, d_4, d_5, d_6\}$. These two records can be linked together because they both contain the BSN; therefore, they are represented by node $A \vDash \{bsn, d_1, \ldots, d_6\}$. Informally, coalitions of honest actors can link records if they have stored a shared identifier; coalitions of curious actors can additionally link records if they have exchanged personal information from the records.

Modelling the optimal situation O-PI makes it possible to assess the extent to which existing infrastructures satisfy data minimisation. Namely, we can build combined coalition graphs comparing existing infrastructures to O-PI, as described in Section 2.4. If the two graphs are the same, the infrastructure achieves optimal privacy. Otherwise, the privacy issues of the analysed infrastructure can be identified by analysing non-optimal nodes in the graph.

8.4 Step 3: Model Communication

The third step of analysing data minimisation using coalition graphs is to model the communication in the systems we consider. We use the model of standard cryptographic primitives from Section 3.4. In addition, we model the pseudocodes used in PS-PI[10] using function symbol pc/3, where $\mathsf{pc}(\mathsf{s}, \mathsf{b}, \mathsf{d})$ represents a pseudocode based on BSN b and domain d using secret s. Its functionality is modelled by the following rules (plus implicit reconstruction rule):

[10] Parelsnoer Initiatief (2009)

$$\mathsf{pc}(s,b,d) \leftarrow s, b, d \quad \mathsf{pc}(s,b,d) \xrightarrow{\doteq s} s \quad \mathsf{pc}(s,b,d) \xrightarrow{\doteq s} b \quad \mathsf{pc}(s,b,d) \xrightarrow{\doteq s} d.$$

Intuitively, pseudocodes can be thought of as deterministic, symmetric encryptions of the BSN and domain using the secret. We model communication in each system as a trace from an initial state $\{\mathcal{C}^0\}_{x \in \mathcal{A}}$ (Section 6.3); the curious actors a^* interact, and send information that should be stored to honest actors a.[11]

[11] For this latter interaction, we do not model communication identifiers.

Our model of communication in H-PI is shown in Figure 8.5. Distribution of patient data to the CI by the two hospitals is modelled by domains α_1 and α_2. Distribution of patient data to the researcher by the CI is modelled by domains β_1 and β_2. The information- and contents-layer model of most pieces of information is obvious; for the domains, $dom|_r^{\beta_1} \doteq dom1|_r^{\cdot} \not\doteq dom2|_r^{\cdot} \doteq dom|_r^{\beta_2}$.

Our model of communication in PS-PI is shown in Figure 8.9 (at the end of this chapter). Compared to our model of H-PI, hospitals now additionally know the domain for their pseudocodes, which they provide to the PS in the first message. Distribution of patient data now involves three parties: the hospital, PS, and CI. Because there is no a priori reason why the PS should know that the requests

$$\mathcal{C}^0_* = \{ip|^{\cdot}_{ttp}, ip|^{\cdot}_{umc1}, ip|^{\cdot}_{umc2}, ip|^{\cdot}_{umc3}, ip|^{\cdot}_{ci}, ip|^{\cdot}_{r}\}$$
$$\mathcal{C}^0_{umc1^*} = \mathcal{C}^0_{umc1} = \mathcal{C}^0_* \cup \{bsn|^{\kappa}_{u}, d_1|^{\kappa}_{u}, d_2|^{\kappa}_{u}, d_3|^{\kappa}_{u}\}$$
$$\mathcal{C}^0_{umc2^*} = \mathcal{C}^0_{umc2} = \mathcal{C}^0_* \cup \{bsn|^{\mu}_{u}, d_4|^{\mu}_{u}, d_5|^{\mu}_{u}, d_6|^{\mu}_{u}\}$$
$$\mathcal{C}^0_{ci^*} = \mathcal{C}^0_{ci} = \mathcal{C}^0_* \cup \{dom1|^{\cdot}_{r}, dom2|^{\cdot}_{r}\}$$
$$\mathcal{C}^0_{r} = \mathcal{C}^0_* \cup \{d_7|^{\beta_1}_{u}\}$$

$$umc1^*(ip|^{\alpha_1}_{umc}) \to ci^*(ip|^{\alpha_1}_{ci}) : \{\mathsf{h}(bsn|^{\alpha_1}_{u}), d_1|^{\alpha_1}_{u}, d_2|^{\alpha_1}_{u}, d_3|^{\alpha_1}_{u}\}$$
$$umc1^* \to umc1 : \{bsn|^{\alpha_1}_{u}, \mathsf{h}(bsn|^{\alpha_1}_{u})\}$$
$$ci^* \to ci : \{\mathsf{h}(bsn|^{\alpha_1}_{u}), ip|^{\alpha_1}_{umc}, d_1|^{\alpha_1}_{u}, d_2|^{\alpha_1}_{u}, d_3|^{\alpha_1}_{u}\}$$
$$umc2^*(ip|^{\alpha_2}_{umc}) \to ci^*(ip|^{\alpha_2}_{ci}) : \{\mathsf{h}(bsn|^{\alpha_2}_{u}), d_4|^{\alpha_2}_{u}, d_5|^{\alpha_2}_{u}, d_6|^{\alpha_2}_{u}\}$$
$$umc2^* \to umc2 : \{bsn|^{\alpha_2}_{u}, \mathsf{h}(bsn|^{\alpha_2}_{u})\}$$
$$ci^* \to ci : \{\mathsf{h}(bsn|^{\alpha_2}_{u}), ip|^{\alpha_2}_{umc}, d_4|^{\alpha_2}_{u}, d_5|^{\alpha_2}_{u}, d_6|^{\alpha_2}_{u}\}$$
$$ci^*(ip|^{\beta_1}_{ci}) \to r(ip|^{\beta_1}_{r}) : \{\mathsf{h}(\{\mathsf{h}(bsn|^{\beta_1}_{u}), dom|^{\beta_1}_{r}\}), d_1|^{\beta_1}_{u}, d_4|^{\beta_1}_{u}\}$$
$$ci^* \to ci : \{\mathsf{h}(bsn|^{\beta_1}_{u}), \mathsf{h}(\{\mathsf{h}(bsn|^{\beta_1}_{u}), dom|^{\beta_1}_{r}\}), dom|^{\beta_1}_{r}, d_1|^{\beta_1}_{u}, d_4|^{\beta_1}_{u}\}$$
$$ci^*(ip|^{\beta_2}_{ci}) \to r(ip|^{\beta_2}_{r}) : \{\mathsf{h}(\{\mathsf{h}(bsn|^{\beta_2}_{u}), dom|^{\beta_2}_{r}\}), d_2|^{\beta_2}_{u}, d_5|^{\beta_2}_{u}\}$$
$$ci^* \to ci : \{\mathsf{h}(bsn|^{\beta_2}_{u}), \mathsf{h}(\{\mathsf{h}(bsn|^{\beta_2}_{u}), dom|^{\beta_2}_{r}\}), dom|^{\beta_2}_{r}, d_2|^{\beta_2}_{u}, d_5|^{\beta_2}_{u}\}$$

Figure 8.5: Model of communication in H-PI: initial state (top) and trace (bottom)

of the hospital and the CI are related, we model these different parts of the distribution process using different domains. For instance, when *umc*1 distributes data to the CI, it first pseudonymises it (domain $\alpha_{1,1}$); then provides it to the CI ($\alpha_{1,2}$) who re-pseudonymises it ($\alpha_{1,3}$); and finally stores it in a database (domain α_1).[12] Distribution of data to the researcher is similar: the CI first repseudonymises the data (domain $\beta_{i,1}$); then provides it to the researcher ($\beta_{i,2}$); and finally archives it (β_i).

[12] Of course, in this case, the PS *can* relate the requests, i.e., it can associate the patient in the different domains. By modelling the three different domains, we obtain this as a result of our analysis rather than by assuming it a priori.

8.5 Step 4: Compare Knowledge

The final step of analysing data minimisation is to compare the knowledge in the different systems using coalition graphs. Namely, our tool automatically produces coalition graphs for each individual system, and combined coalition graphs comparing the knowledge from different systems. Using these graphs, we can analyse data minimisation and suggest privacy improvements.

Hash-Based Infrastructure Figure 8.6 shows the combined coalition graph of the hash-based Parelsnoer infrastructure (H-PI) and the optimal situation (O-PI). The dotted nodes represent nodes that only occur in H-PI's coalition graph and thus point to violations of data minimisation. The solid nodes are also in O-PI's coalition graph and thus assumed unavoidable.[13]

The non-optimal nodes can be explained by the use of translation tables for depseudonymisation, as opposed to using the services

[13] H-PI does not use the TTP; it occurs in this graph because we compare it to the optimal situation, in which the TTP is needed for (de)pseudonymisation.

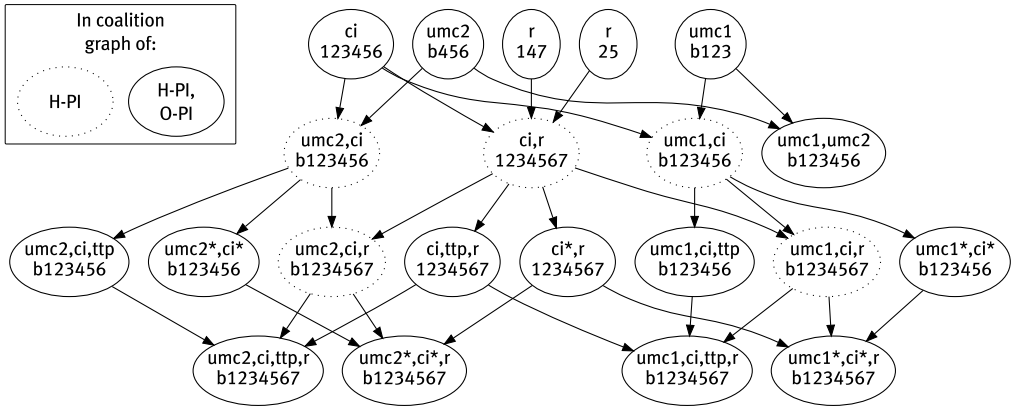


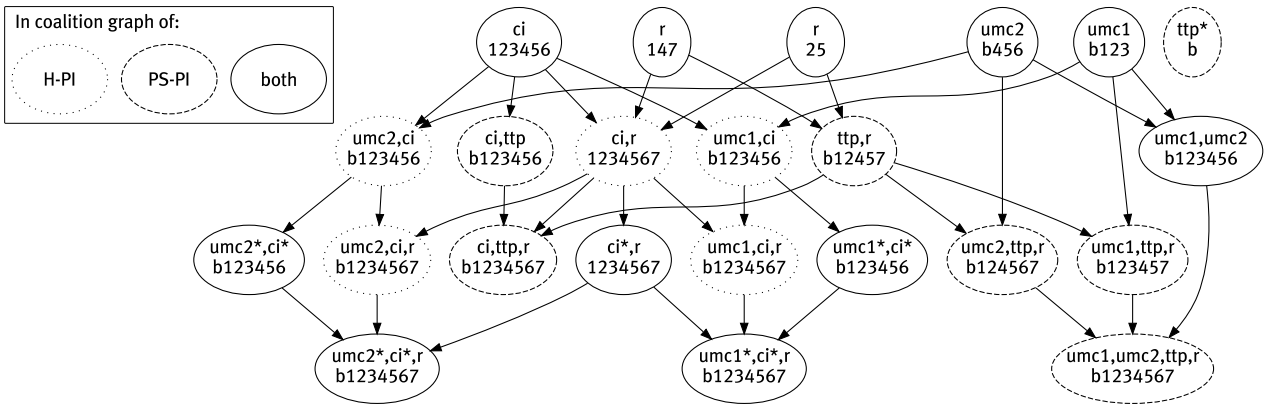
Figure 8.6: Combined coalition graph of the Parelsnoer hash-based pseudonymisation infrastructure (H-PI) and optimal situation (O-PI)

of the TTP. Hospitals need to remember the pseudocode sent to the CI for full depseudonymisation, which implies $\{umc_i, ci\} \vDash \{bsn, d_1, \ldots, d_6\}$. The CI needs to remember the pseudocode sent to the researcher, implying $\{ci, r\} \vDash \{d_1, \ldots, d_7\}$. Combining the translation tables gives $\{umc_i, ci, r\} \vDash \{bsn, d_1, \ldots, d_7\}$. Note that, for any non-optimal node $A \vDash O$ occurring in H-PI's graph, node $A \cup \{ttp\} \vDash O$ is optimal. This expresses that actors A should be allowed to compile record O, but only through a rigorous process involving the TTP. Also, for any non-optimal node $A \vDash O$, node $A' \vDash O$ is optimal in which hospitals and CI in A are curious. This means that these actors store more data than is desirable. It also means that, if all actors are assumed to be curious, then H-PI is optimal.

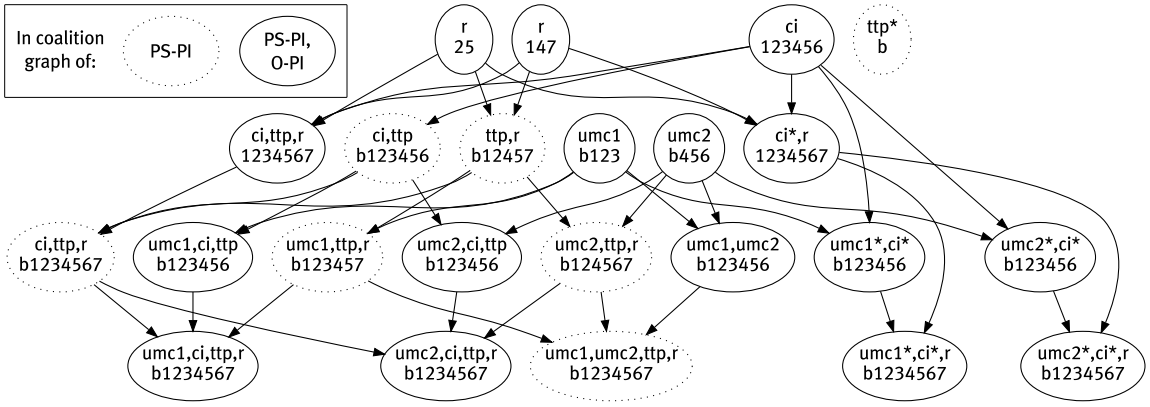
Pseudonymisation Service We now discuss privacy in the Pseudonymisation Service infrastructure PS-PI. We compare it to the hash-based infrastructure (Figure 8.7(a)) and to the optimal situation (Figure 8.7(b)).

Figure 8.7(a) shows that all non-optimal nodes of H-PI (shown dotted) are eliminated in PS-PI; however, PS-PI introduces new non-optimal nodes (shown dashed) which reflect two new privacy problems. The first problem is that the PS ttp learns the patient's BSN in the pseudonymisation process, and can contribute this information to coalitions that should not know it. This is reflected by nodes $\{ttp^*\} \vDash \{bsn\}$, $\{ci, ttp\} \vDash \{bsn, d_1, \ldots, d_6\}$, and $\{ci, ttp, r\} \vDash \{bsn, d_1, \ldots, d_7\}$ (in H-PI, these actors know the same data, but without the BSN). The second problem is that the PS is able to link records held by researchers and hospitals without involving the CI. This problem, combined with the first problem, is reflected by nodes $\{ttp, r\} \vDash \{bsn, d_1, d_2, d_4, d_5, d_7\}$ (linking records from different research projects); $\{umc_1, ttp, r\} \vDash \{bsn, d_1, d_2, d_3, d_4, d_5, d_7\}$ and $\{umc_2, ttp, r, \} \vDash \{bsn, d_1, d_2, d_4, d_5, d_6, d_7\}$ (linking records from researcher and hospital); and $\{umc_1, umc_2, ttp, r\} \vDash \{bsn, d_1, \ldots, d_7\}$ (combination of the two). As Figure 8.7(b) shows, these nodes, which all include the PS, are exactly PS-PI's non-optimal nodes.

The analysis shows how privacy protection in PS-PI crucially



(a) Combined coalition graph of PS-PI and H-PI



(b) Combined coalition graph of PS-PI and O-PI

Figure 8.7: Comparison of the Pseudonymisation Service infrastructure (PS-PI) with the hash-based infrastructure (H-PI) and the optimal situation (O-PI)

depends on the trustworthiness of the PS. If we assume that the PS is never involved in privacy breaches, then coalitions including the PS are not relevant; in this case, PS-PI is optimal. However, without this assumption, PS-PI provides worse privacy than H-PI by offering additional ways to establish links and find out the patient's BSN. In particular, a curious PS can find out the BSN, which is actually forbidden by Dutch legislation. To mitigate this, measures should be taken to make sure that the PS cannot use the BSN, e.g., by carrying out all computations on the BSN using trusted hardware (as done by Parelsnoer).

8.6 From PS-PI to an Optimal System

In the previous section, we have identified two privacy issues in the PS-PI infrastructure. We now discuss solutions, and then consider a hypothetical infrastructure incorporating these solutions and analyse it using coalition graphs.

The first privacy problem is that the PS learns the patient's BSN. Although it may be mitigated using trusted hardware, it is desirable to technically ensure that the BSN does not leave the hospitals. The main challenge in achieving this is that the CI needs to link records from different hospitals. In particular, all hospitals should use the same pseudonym of a patient when communicating with the PS. In-

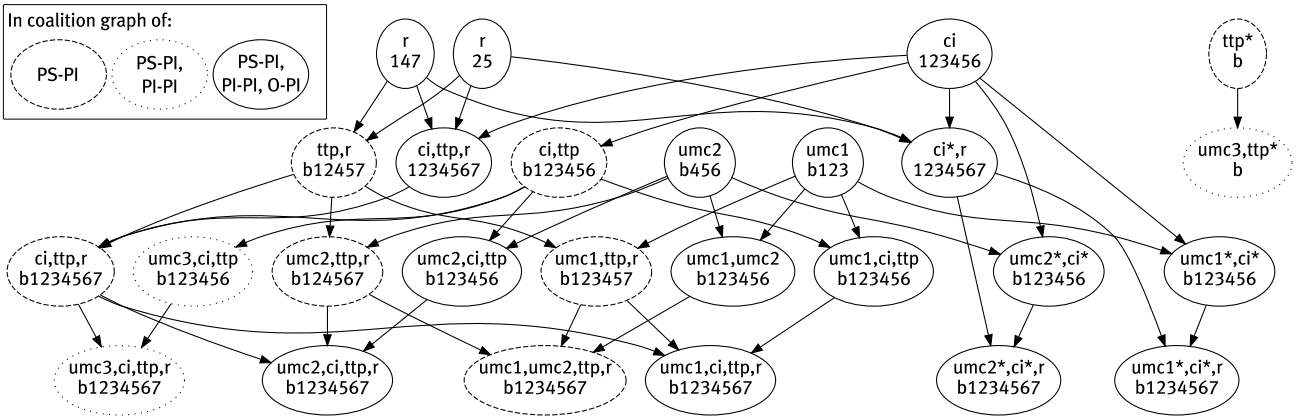


Figure 8.8: Comparison of reduced coalition graphs of the improved PS infrastructure (PI-PI) with the original PS infrastructure (PS-PI) and the optimal situation (O-PI)

tuitively, all hospitals should use a shared secret to generate pseudonyms, or in case they do not share any secret, they should use the same procedure to generate pseudonyms, for instance hashing BSNs as in H-PI. The drawback of the first solution is that depseudonymisation can also be performed by hospitals that do not have a record of the patient (e.g., umc_3 in our scenario). On the other hand, if the pseudonyms are generated using one procedure, they may be vulnerable to dictionary attacks as in H-PI. We leave further analysis of this issue as future work.

The second problem is that the PS can help researchers link their data to hospitals or other researchers, bypassing the CI. To solve this, the PS should not be able to link pseudonymisation requests for different domains. This means that when the CI compiles a dataset for distribution, it should modify its linkable pseudocode before requesting the PS to repseudonymise it. The CI may either use the same secret for all datasets, or use different secrets for different datasets or records: both approaches seem possible.

To evaluate the privacy impact of the discussed solutions on PS-PI, we analyse an infrastructure PI-PI that incorporates the solutions in PS-PI. To make sure the BSN does not leave the hospitals, all hospitals share a symmetric key; instead of providing the BSN to the PS, they provide an encryption of the BSN under this key. To prevent linking of distributed datasets by the PS, the CI has a symmetric key for each research domain; when compiling a dataset for distribution, it sends to the PS not his pseudocode itself, but an encryption of the pseudocode under this symmetric key. Instead of re-translating the pseudocode from the CI's domain, the PS simply constructs a new pseudocode using this encryption as pseudonym.

Our model of communication in PI-PI is shown in Figure 8.10 (at the end of this chapter). The communication steps are the same as in PS-PI, but the message formats are changed to reflect the changes described above. The remarks above about our model of PS-PI also apply here. Note that also $umc3$ knows the hospital-shared secret $s_{umc}|_{\cdot}$.[14]

Figure 8.8 compares PI-PI with the original infrastructure PS-

[14] In our formalisation, ci^* is assumed to initially know the pseudocode in domains $\beta_{1,1}$ and $\beta_{2,1}$. This is technically needed to ensure validity of the trace. Namely, because ttp^* initially knows the secret in the pseudocode, ci^* needs to know that the pseudocode he sends respects this secret. The only way to model this knowledge without giving ci^* the secret is by adding the pseudocode to his initial knowledge.

PI and the optimal situation O-PI. As the figure shows, PI-PI indeed solves the privacy problems in PS-PI; however, one problem remains. Namely, besides umc_1 and umc_2, umc_3 can also help in depseudonymisation although it does not know the patient. Note that H-PI does not have this problem because umc_3 does not know the BSN and birth date of the patient. Hence, the privacy of H-PI and PI-PI is formally incomparable. In practice, we have a choice between depseudonymisation by any hospital knowing a secret (PI-PI), or by any third party able to perform a dictionary attack (H-PI).

8.7 Discussion

This analysis previously appeared in Veeningen et al. (2012).

The above analysis of data minimisation of (de)pseudonymisation systems is only with respect to the assumed design decisions. In particular, we assume a central infrastructure that is allowed to learn the attributes about all patients, as long as they remain anonymised. Hence, possible approaches to prevent this knowledge, e.g., using cryptographic techniques, are not considered. In particular, as an example of such techniques we mention the proposal by Quantin et al.[15] to use secret sharing to divide patient information between two parties that separately do not learn any information.

[15] Quantin et al. (2011)

Apart from Parelsnoer and the proposal by Quantin et al., several other proposals for (de)-pseudonymisation of patient data for medical research exist. Pommerening and Reng[16] consider several designs in the German legal framework that are similar to H-PI and PS-PI, so we expect the findings of our analysis to also apply there. Claerhout and De Moor[17] report on a Belgian model that, instead of central storage, uses a pseudonymisation service that also distributes the data (though encrypted so that the PS cannot read it). More general approaches for the exchange of medical data between health care providers[18] or pseudonymised data in general[19] may also be usable for pseudonymisation for research purposes. While some general discussions on privacy aspects of (de)pseudonymisation in the medical context exist[20], we are not aware of any comprehensive comparisons of the privacy characteristics of the above systems. Generalising our present analysis to include these systems is an interesting direction for future work.

[16] Pommerening and Reng (2004)

[17] Claerhout and De Moor (2005)

[18] E.g., Deng et al. (2009), Riedl et al. (2007), Zhang et al. (2005)

[19] E.g., Teepe (2005)

[20] E.g., Tinabo et al. (2009), Office of the Data Protection Commissioner (Ireland) (2007)

$$
\begin{aligned}
\mathcal{C}^0_{*} &= \{ip|^{\cdot}_{ttp}, ip|^{\cdot}_{umc1}, ip|^{\cdot}_{umc2}, ip|^{\cdot}_{umc3}, ip|^{\cdot}_{ci}, ip|^{\cdot}_{r}\}\\
\mathcal{C}^0_{umc1^*} &= \mathcal{C}^0_{umc1} = \mathcal{C}^0_{*} \cup \{bsn|^{\kappa}_{u}, d_1|^{\kappa}_{u}, d_2|^{\kappa}_{u}, d_3|^{\kappa}_{u}, dom|^{\cdot}_{umc1}\}\\
\mathcal{C}^0_{umc2^*} &= \mathcal{C}^0_{umc2} = \mathcal{C}^0_{*} \cup \{bsn|^{\mu}_{u}, d_4|^{\mu}_{u}, d_5|^{\mu}_{u}, d_6|^{\mu}_{u}, dom|^{\cdot}_{umc2}\}\\
\mathcal{C}^0_{ci^*} &= \mathcal{C}^0_{ci} = \mathcal{C}^0_{*} \cup \{dom1|^{\cdot}_{r}, dom2|^{\cdot}_{r}, dom|^{\cdot}_{ci}\}\\
\mathcal{C}^0_{r} &= \mathcal{C}^0_{*} \cup \{d_7|^{\beta_{1,2}}_{u}\}\\
\mathcal{C}^0_{ttp^*} &= \mathcal{C}^0_{ttp} = \mathcal{C}^0_{*} \cup \{s|^{\cdot}, dom|^{\cdot}_{umc1}, dom|^{\cdot}_{umc2}, dom|^{\cdot}_{umc3}, dom|^{\cdot}_{ci}\}
\end{aligned}
$$

$$
\begin{aligned}
umc1^*(ip|^{\alpha_{1,1}}_{umc}) \rightarrow ttp^*(ip|^{\alpha_{1,1}}_{ttp}) &: \{bsn|^{\alpha_{1,1}}_{u}, dom|^{\alpha_{1,1}}_{umc}\}\\
ttp^*(ip|^{\alpha_{1,1}}_{ttp}) \rightarrow umc1^*(ip|^{\alpha_{1,1}}_{umc}) &: \mathsf{pc}(s|^{\alpha_{1,1}}, bsn|^{\alpha_{1,1}}_{u}, dom|^{\alpha_{1,1}}_{umc})\\
umc1^*(ip|^{\alpha_{1,2}}_{umc}) \rightarrow ci^*(ip|^{\alpha_{1,2}}_{ci}) &: \{\mathsf{pc}(s|^{\alpha_{1,2}}, bsn|^{\alpha_{1,2}}_{u}, dom|^{\alpha_{1,2}}_{umc}), dom|^{\alpha_{1,2}}_{umc}, d_1|^{\alpha_{1,2}}_{u}, d_2|^{\alpha_{1,2}}_{u}, d_3|^{\alpha_{1,2}}_{u}\}\\
ci^*(ip|^{\alpha_{1,3}}_{ci}) \rightarrow ttp^*(ip|^{\alpha_{1,3}}_{ttp}) &: \{\mathsf{pc}(s|^{\alpha_{1,3}}, bsn|^{\alpha_{1,3}}_{u}, dom|^{\alpha_{1,3}}_{umc}), dom|^{\alpha_{1,3}}_{umc}, dom|^{\alpha_{1,3}}_{ci}\}\\
ttp^*(ip|^{\alpha_{1,3}}_{ttp}) \rightarrow ci^*(ip|^{\alpha_{1,3}}_{ci}) &: \mathsf{pc}(s|^{\alpha_{1,3}}, bsn|^{\alpha_{1,3}}_{u}, dom|^{\alpha_{1,3}}_{ci})\\
ci^* \rightarrow ci &: \{ip|^{\alpha_1}_{umc}, dom|^{\alpha_1}_{umc}, d_1|^{\alpha_1}_{u}, d_2|^{\alpha_1}_{u}, d_3|^{\alpha_1}_{u}, \mathsf{pc}(s|^{\alpha_1}, bsn|^{\alpha_1}_{u}, dom|^{\alpha_1}_{ci}), dom|^{\alpha_1}_{ci}\}\\
umc2^*(ip|^{\alpha_{2,1}}_{umc}) \rightarrow ttp^*(ip|^{\alpha_{2,1}}_{ttp}) &: \{bsn|^{\alpha_{2,1}}_{u}, dom|^{\alpha_{2,1}}_{umc}\}\\
ttp^*(ip|^{\alpha_{2,1}}_{ttp}) \rightarrow umc2^*(ip|^{\alpha_{2,1}}_{umc}) &: \mathsf{pc}(s|^{\alpha_{2,1}}, bsn|^{\alpha_{2,1}}_{u}, dom|^{\alpha_{2,1}}_{umc})\\
umc2^*(ip|^{\alpha_{2,2}}_{umc}) \rightarrow ci^*(ip|^{\alpha_{2,2}}_{ci}) &: \{\mathsf{pc}(s|^{\alpha_{2,2}}, bsn|^{\alpha_{2,2}}_{u}, dom|^{\alpha_{2,2}}_{umc}), dom|^{\alpha_{2,2}}_{umc}, d_4|^{\alpha_{2,2}}_{u}, d_5|^{\alpha_{2,2}}_{u}, d_6|^{\alpha_{2,2}}_{u}\}\\
ci^*(ip|^{\alpha_{2,3}}_{ci}) \rightarrow ttp^*(ip|^{\alpha_{2,3}}_{ttp}) &: \{\mathsf{pc}(s|^{\alpha_{2,3}}, bsn|^{\alpha_{2,3}}_{u}, dom|^{\alpha_{2,3}}_{umc}), dom|^{\alpha_{2,3}}_{umc}, dom|^{\alpha_{2,3}}_{ci}\}\\
ttp^*(ip|^{\alpha_{2,3}}_{ttp}) \rightarrow ci^*(ip|^{\alpha_{2,3}}_{ci}) &: \mathsf{pc}(s|^{\alpha_{2,3}}, bsn|^{\alpha_{2,3}}_{u}, dom|^{\alpha_{2,3}}_{ci})\\
ci^* \rightarrow ci &: \{ip|^{\alpha_2}_{umc}, dom|^{\alpha_2}_{umc}, d_4|^{\alpha_2}_{u}, d_5|^{\alpha_2}_{u}, d_6|^{\alpha_2}_{u}, \mathsf{pc}(s|^{\alpha_2}, bsn|^{\alpha_2}_{u}, dom|^{\alpha_2}_{ci}), dom|^{\alpha_2}_{ci}, dom|^{\alpha_2}_{umc}\}\\
ci^*(ip|^{\beta_{1,1}}_{ci}) \rightarrow ttp^*(ip|^{\beta_{1,1}}_{ttp}) &: \{\mathsf{pc}(s|^{\beta_{1,1}}, bsn|^{\beta_{1,1}}_{u}, dom|^{\beta_{1,1}}_{ci}), dom|^{\beta_{1,1}}_{r}\}\\
ttp^*(ip|^{\beta_{1,1}}_{ttp}) \rightarrow ci^*(ip|^{\beta_{1,1}}_{ci}) &: \mathsf{pc}(s|^{\beta_{1,1}}, bsn|^{\beta_{1,1}}_{u}, dom|^{\beta_{1,1}}_{r})\\
ci^*(ip|^{\beta_{1,2}}_{ci}) \rightarrow r(ip|^{\beta_{1,2}}_{r}) &: \{\mathsf{pc}(s|^{\beta_{1,2}}, bsn|^{\beta_{1,2}}_{u}, dom|^{\beta_{1,2}}_{r}), d_1|^{\beta_{1,2}}_{u}, d_4|^{\beta_{1,2}}_{u}\}\\
ci^* \rightarrow ci &: \{\mathsf{pc}(s|^{\beta_1}, bsn|^{\beta_1}_{u}, dom|^{\beta_1}_{r}), dom|^{\beta_1}_{r}\}\\
ci^*(ip|^{\beta_{2,2}}_{ci}) \rightarrow ttp^*(ip|^{\beta_{2,2}}_{ttp}) &: \{\mathsf{pc}(s|^{\beta_{2,2}}, bsn|^{\beta_{2,2}}_{u}, dom|^{\beta_{2,2}}_{ci}), dom|^{\beta_{2,2}}_{r}\}\\
ttp^*(ip|^{\beta_{2,2}}_{ttp}) \rightarrow ci^*(ip|^{\beta_{2,2}}_{ci}) &: \mathsf{pc}(s|^{\beta_{2,2}}, bsn|^{\beta_{2,2}}_{u}, dom|^{\beta_{2,2}}_{r})\\
ci^*(ip|^{\beta_{2,2}}_{ci}) \rightarrow r(ip|^{\beta_{2,2}}_{r}) &: \{\mathsf{pc}(s|^{\beta_{2,2}}, bsn|^{\beta_{2,2}}_{u}, dom|^{\beta_{2,2}}_{r}), d_2|^{\beta_{2,2}}_{u}, d_5|^{\beta_{2,2}}_{u}\}\\
ci^* \rightarrow ci &: \{\mathsf{pc}(s|^{\beta_2}, bsn|^{\beta_2}_{u}, dom|^{\beta_2}_{r}), dom|^{\beta_2}_{r}\}
\end{aligned}
$$

Figure 8.9: Model of communication in PS-PI: initial state (top) and trace (bottom)

$$
\begin{aligned}
\mathcal{C}^0_{*} &= \{ip|^{\cdot}_{ttp}, ip|^{\cdot}_{umc1}, ip|^{\cdot}_{umc2}, ip|^{\cdot}_{umc3}, ip|^{\cdot}_{ci}, ip|^{\cdot}_{r}\}\\
\mathcal{C}^0_{umc1*} = \mathcal{C}^0_{umc1} &= \mathcal{C}^0_{*} \cup \{bsn|^{\kappa}_{u}, d_1|^{\kappa}_{u}, d_2|^{\kappa}_{u}, d_3|^{\kappa}_{u}, dom|^{\cdot}_{umc1}, dom|^{\cdot}_{umc1}, s_{umc}|^{\cdot}_{\cdot}\}\\
\mathcal{C}^0_{umc2*} = \mathcal{C}^0_{umc2} &= \mathcal{C}^0_{*} \cup \{bsn|^{\mu}_{u}, d_4|^{\mu}_{u}, d_5|^{\mu}_{u}, d_6|^{\mu}_{u}, dom|^{\cdot}_{umc2}, dom|^{\cdot}_{umc2}, s_{umc}|^{\cdot}_{\cdot}\}\\
\mathcal{C}^0_{umc3*} = \mathcal{C}^0_{umc3} &= \mathcal{C}^0_{*} \cup \{s_{umc}|^{\cdot}_{\cdot}\}\\
\mathcal{C}^0_{ci} &= \mathcal{C}^0_{*} \cup \{dom|^{\cdot}_{ci}, dom_1|^{\cdot}_{r}, dom_2|^{\cdot}_{r}, sr_1|^{\cdot}_{r}, sr_2|^{\cdot}_{r}\}\\
\mathcal{C}^0_{ci*} &= \mathcal{C}^0_{ci} \cup \{\mathsf{pc}(s|^{\beta_{1,1}}_{\cdot}, \mathsf{enc}(bsn|^{\beta_{1,1}}_{u}, s_{umc}|^{\beta_{1,1}}_{\cdot}), dom|^{\beta_{1,1}}_{ci}), \mathsf{pc}(s|^{\beta_{2,1}}_{\cdot}, \mathsf{enc}(bsn|^{\beta_{2,1}}_{u}, s_{umc}|^{\beta_{2,1}}_{\cdot}), dom|^{\beta_{2,1}}_{ci})\}\\
\mathcal{C}^0_{r} &= \mathcal{C}^0_{*} \cup \{d_7|^{\beta_{1,2}}_{u}\}\\
\mathcal{C}^0_{ttp*} = \mathcal{C}^0_{ttp} &= \mathcal{C}^0_{*} \cup \{s|^{\cdot}_{\cdot}, s|^{\beta_{1,1}}_{\cdot}, s|^{\beta_{2,1}}_{\cdot}, dom|^{\cdot}_{umc1}, dom|^{\cdot}_{umc2}, dom|^{\cdot}_{umc3}, dom|^{\cdot}_{ci}\}
\end{aligned}
$$

$$
\begin{aligned}
umc1^*(ip|^{\alpha_{1,1}}_{umc}) \rightarrow ttp^*(ip|^{\alpha_{1,1}}_{ttp}) &: \{\mathsf{enc}(bsn|^{\alpha_{1,1}}_{u}, s_{umc}|^{\alpha_{1,1}}_{\cdot}), dom|^{\alpha_{1,1}}_{umc}\}\\
ttp^*(ip|^{\alpha_{1,1}}_{ttp}) \rightarrow umc1^*(ip|^{\alpha_{1,1}}_{umc}) &: \mathsf{pc}(s|^{\alpha_{1,1}}_{\cdot}, \mathsf{enc}(bsn|^{\alpha_{1,1}}_{u}, s_{umc}|^{\alpha_{1,1}}_{\cdot}), dom|^{\alpha_{1,1}}_{umc})\\
umc1^*(ip|^{\alpha_{1,2}}_{umc}) \rightarrow ci^*(ip|^{\alpha_{1,2}}_{ci}) &: \{\mathsf{pc}(s|^{\alpha_{1,2}}_{\cdot}, \mathsf{enc}(bsn|^{\alpha_{1,2}}_{u}, s_{umc}|^{\alpha_{1,2}}_{\cdot}), dom|^{\alpha_{1,2}}_{umc}), dom|^{\alpha_{1,2}}_{umc}, d_1|^{\alpha_{1,2}}_{u}, d_2|^{\alpha_{1,2}}_{u}, d_3|^{\alpha_{1,2}}_{u}\}\\
ci^*(ip|^{\alpha_{1,3}}_{ci}) \rightarrow ttp^*(ip|^{\alpha_{1,3}}_{ttp}) &: \{\mathsf{pc}(s|^{\alpha_{1,3}}_{\cdot}, \mathsf{enc}(bsn|^{\alpha_{1,3}}_{u}, s_{umc}|^{\alpha_{1,3}}_{\cdot}), dom|^{\alpha_{1,3}}_{umc}), dom|^{\alpha_{1,3}}_{umc}, dom|^{\alpha_{1,3}}_{ci}\}\\
ttp^*(ip|^{\alpha_{1,3}}_{ttp}) \rightarrow ci^*(ip|^{\alpha_{1,3}}_{ci}) &: \mathsf{pc}(s|^{\alpha_{1,3}}_{\cdot}, \mathsf{enc}(bsn|^{\alpha_{1,3}}_{u}, s_{umc}|^{\alpha_{1,3}}_{\cdot}), dom|^{\alpha_{1,3}}_{ci})\\
ci^* \rightarrow ci &: \{\mathsf{pc}(s|^{\alpha_1}_{\cdot}, \mathsf{enc}(bsn|^{\alpha_1}_{u}, s_{umc}|^{\alpha_1}_{\cdot}), dom|^{\alpha_1}_{ci}), dom|^{\alpha_1}_{ci}, ip|^{\alpha_1}_{umc}, dom|^{\alpha_1}_{umc}, d_1|^{\alpha_1}_{u}, d_2|^{\alpha_1}_{u}, d_3|^{\alpha_1}_{u}\}\\
umc2^*(ip|^{\alpha_{2,1}}_{umc}) \rightarrow ttp^*(ip|^{\alpha_{2,1}}_{ttp}) &: \{\mathsf{enc}(bsn|^{\alpha_{2,1}}_{u}, s_{umc}|^{\alpha_{2,1}}_{\cdot}), dom|^{\alpha_{2,1}}_{umc}\}\\
ttp^*(ip|^{\alpha_{2,1}}_{ttp}) \rightarrow umc2^*(ip|^{\alpha_{2,1}}_{umc}) &: \mathsf{pc}(s|^{\alpha_{2,1}}_{\cdot}, \mathsf{enc}(bsn|^{\alpha_{2,1}}_{u}, s_{umc}|^{\alpha_{2,1}}_{\cdot}), dom|^{\alpha_{2,1}}_{umc})\\
umc2^*(ip|^{\alpha_{2,2}}_{umc}) \rightarrow ci^*(ip|^{\alpha_{2,2}}_{ci}) &: \{\mathsf{pc}(s|^{\alpha_{2,2}}_{\cdot}, \mathsf{enc}(bsn|^{\alpha_{2,2}}_{u}, s_{umc}|^{\alpha_{2,2}}_{\cdot}), dom|^{\alpha_{2,2}}_{umc}), dom|^{\alpha_{2,2}}_{umc}, d_4|^{\alpha_{2,2}}_{u}, d_5|^{\alpha_{2,2}}_{u}, d_6|^{\alpha_{2,2}}_{u}\}\\
ci^*(ip|^{\alpha_{2,3}}_{ci}) \rightarrow ttp^*(ip|^{\alpha_{2,3}}_{ttp}) &: \{\mathsf{pc}(s|^{\alpha_{2,3}}_{\cdot}, \mathsf{enc}(bsn|^{\alpha_{2,3}}_{u}, s_{umc}|^{\alpha_{2,3}}_{\cdot}), dom|^{\alpha_{2,3}}_{umc}), dom|^{\alpha_{2,3}}_{umc}, dom|^{\alpha_{2,3}}_{ci}\}\\
ttp^*(ip|^{\alpha_{2,3}}_{ttp}) \rightarrow ci^*(ip|^{\alpha_{2,3}}_{ci}) &: \mathsf{pc}(s|^{\alpha_{2,3}}_{\cdot}, \mathsf{enc}(bsn|^{\alpha_{2,3}}_{u}, s_{umc}|^{\alpha_{2,3}}_{\cdot}), dom|^{\alpha_{2,3}}_{ci})\\
ci^* \rightarrow ci &: \{\mathsf{pc}(s|^{\alpha_2}_{\cdot}, \mathsf{enc}(bsn|^{\alpha_2}_{u}, s_{umc}|^{\alpha_2}_{\cdot}), dom|^{\alpha_2}_{ci}), dom|^{\alpha_2}_{ci}, ip|^{\alpha_2}_{umc}, dom|^{\alpha_2}_{umc}, d_4|^{\alpha_2}_{u}, d_5|^{\alpha_2}_{u}, d_6|^{\alpha_2}_{u}\}\\
ci^*(ip|^{\beta_{1,1}}_{ci}) \rightarrow ttp^*(ip|^{\beta_{1,1}}_{ttp}) &: \{\mathsf{enc}(\mathsf{pc}(s|^{\beta_{1,1}}_{\cdot}, \mathsf{enc}(bsn|^{\beta_{1,1}}_{u}, s_{umc}|^{\beta_{1,1}}_{\cdot}), dom|^{\beta_{1,1}}_{ci}), sr|^{\beta_{1,1}}_{r}), dom|^{\beta_{1,1}}_{r}\}\\
ttp^*(ip|^{\beta_{1,1}}_{ttp}) \rightarrow ci^*(ip|^{\beta_{1,1}}_{ci}) &: \mathsf{pc}(s|^{\beta_{1,1}}_{\cdot}, \mathsf{enc}(\mathsf{pc}(s|^{\beta_{1,1}}_{\cdot}, \mathsf{enc}(bsn|^{\beta_{1,1}}_{u}, s_{umc}|^{\beta_{1,1}}_{\cdot}), dom|^{\beta_{1,1}}_{ci}), sr|^{\beta_{1,1}}_{r}), dom|^{\beta_{1,1}}_{r})\\
ci^*(ip|^{\beta_{1,2}}_{ci}) \rightarrow r^*(ip|^{\beta_{1,2}}_{r}) &: \{\mathsf{pc}(s|^{\beta_{1,2}}_{\cdot}, \mathsf{enc}(\mathsf{pc}(s|^{\beta_{1,2}}_{\cdot}, \mathsf{enc}(bsn|^{\beta_{1,2}}_{u}, s_{umc}|^{\beta_{1,2}}_{\cdot}), dom|^{\beta_{1,2}}_{ci}), sr|^{\beta_{1,2}}_{r}), dom|^{\beta_{1,2}}_{r}), d_1|^{\beta_{1,2}}_{u}, d_4|^{\beta_{1,2}}_{u}\}\\
ci^* \rightarrow ci &: \{\mathsf{pc}(s|^{\beta_1}_{\cdot}, \mathsf{enc}(\mathsf{pc}(s|^{\beta_1}_{\cdot}, \mathsf{enc}(bsn|^{\beta_1}_{u}, s_{umc}|^{\beta_1}_{\cdot}), dom|^{\beta_1}_{ci}), sr|^{\beta_1}_{r}), dom|^{\beta_1}_{r}), dom|^{\beta_1}_{r}\}\\
ci^*(ip|^{\beta_{2,1}}_{ci}) \rightarrow ttp^*(ip|^{\beta_{2,1}}_{ttp}) &: \{\mathsf{enc}(\mathsf{pc}(s|^{\beta_{2,1}}_{\cdot}, \mathsf{enc}(bsn|^{\beta_{2,1}}_{u}, s_{umc}|^{\beta_{2,1}}_{\cdot}), dom|^{\beta_{2,1}}_{ci}), sr|^{\beta_{2,1}}_{r}), dom|^{\beta_{2,1}}_{r}\}\\
ttp^*(ip|^{\beta_{2,1}}_{ttp}) \rightarrow c^*i(ip|^{\beta_{2,1}}_{ci}) &: \mathsf{pc}(s|^{\beta_{2,1}}_{\cdot}, \mathsf{enc}(\mathsf{pc}(s|^{\beta_{2,1}}_{\cdot}, \mathsf{enc}(bsn|^{\beta_{2,1}}_{u}, s_{umc}|^{\beta_{2,1}}_{\cdot}), dom|^{\beta_{2,1}}_{ci}), sr|^{\beta_{2,1}}_{r}), dom|^{\beta_{2,1}}_{r})\\
ci^*(ip|^{\beta_{2,2}}_{ci}) \rightarrow r^*(ip|^{\beta_{2,2}}_{r}) &: \{\mathsf{pc}(s|^{\beta_{2,2}}_{\cdot}, \mathsf{enc}(\mathsf{pc}(s|^{\beta_{2,2}}_{\cdot}, \mathsf{enc}(bsn|^{\beta_{2,2}}_{u}, s_{umc}|^{\beta_{2,2}}_{\cdot}), dom|^{\beta_{2,2}}_{ci}), sr|^{\beta_{2,2}}_{r}), dom|^{\beta_{2,2}}_{r}), d_2|^{\beta_{2,2}}_{u}, d_5|^{\beta_{2,2}}_{u}\}\\
ci^* \rightarrow ci &: \{\mathsf{pc}(s|^{\beta_2}_{\cdot}, \mathsf{enc}(\mathsf{pc}(s|^{\beta_2}_{\cdot}, \mathsf{enc}(bsn|^{\beta_2}_{u}, s_{umc}|^{\beta_2}_{\cdot}), dom|^{\beta_2}_{ci}), sr|^{\beta_2}_{r}), dom|^{\beta_2}_{r}), dom|^{\beta_2}_{r}\}
\end{aligned}
$$

Figure 8.10: Model of communication in PI-PI: initial state (top) and trace (bottom)

9
Related Work

Contents

IN THIS THESIS, we have presented methods to formally analyse what privacy-sensitive information can be derived from communication protocols. As discussed in Chapter 1, various other approaches exist to achieve this purpose. In this Chapter, we discuss the main other approaches, and compare them to our framework.

Outline In this chapter:

- We discuss existing formal methods approaches for analysing knowledge of actors in communication protocols (§9.1);
- We discuss privacy properties defined in the literature, and what approaches have been used to verify them (§9.2);
- We present a more detailed comparison between our work and popular definitions of privacy properties using equivalences (§9.3);
- We mention alternatives to privacy property verification (§9.4).

9.1 Protocol Analysis

We identify two main approaches for analysing the knowledge of actors in communication protocols: *state-based* and *equivalence-based*.

In *state-based* approaches, desired properties about the knowledge of actors are defined in terms of evolutions of a single instance of a system. For instance, a piece of information is secret if, in all possible evolutions of the system, the attacker does not have enough knowledge to derive it. The possible system evolutions can be modelled using process algebras[1] or other approaches, e.g., induction[2]. The knowledge of an attacker is then analysed based on the set of

[1] E.g., Abadi and Fournet (2001), Boreale (2001), Milner (1999)
[2] Paulson (1998)

messages he has observed during a system evolution. This knowledge analysis is at the core of any state-based protocol verification technique, because it determines what next states are reachable. Message analysis was historically done using deductive systems[3], but more recent formalisms[4] also allow equational theories. Available automated verification tools for state-based analysis include AVISPA[5], ProVerif[6], and Tamarin[7]. Alternatively, verification can be performed semi-manually using theorem-proving tools with the inductive method[8].

[3] Dolev and Yao (1981); Clarke et al. (1998); and Boreale (2001)
[4] Abadi and Fournet (2001); and Armando et al. (2005)
[5] Armando et al. (2005)
[6] Blanchet and Smyth (2011)
[7] Schmidt et al. (2012)
[8] Paulson (1998)

In the second, *equivalence-based* approach, desired properties about the knowledge of actors are defined by comparing the evolutions of two instances of a system. Intuitively, two instances of a system are "equivalent" if, from the outside, it is impossible to distinguish the two instances. Traditionally, in model checking, equivalences are used to show that one system correctly implements another, more abstract system[9]. In protocol verification, because "telling instances apart" is with respect to the knowledge of an attacker, equivalences can also be used to define privacy properties. Namely, this is done by verifying equivalence of two system instances that coincide on public information (e.g., in e-voting: the number of votes cast for each candidate) but differ on information that should remain private (e.g., who voted for which candidate). In more detail, equivalence of two system instances means that for any evolution of the first instance, there should be an evolution of the second instance in which the sets of messages observed by the attacker are "similar". This similarity, formalised as "static equivalence" (cf. Chapter 4) is usually defined using equational theories, but can also be defined using deductive techniques. For automatically verifying equivalences, ProVerif[10] is the main available tool. As mentioned in the introduction, this tool has the disadvantage that it cannot verify all equivalences: in particular, it applies an over-approximation that fails to verify many equivalences; and in many cases, it does not terminate.

[9] Baier and Katoen (2008)
[10] Blanchet and Smyth (2011)

The main difference between the state-based and equivalence-based approach is that the latter one also takes "implicit flows"[11] into account. Namely, state-based approaches do not capture the situation when a privacy-sensitive piece of information is not transmitted itself, but does influence other messages. For instance, if a "yes" vote leads to the transmission of a signature for public key pk_1 and a "no" vote leads to a signature for public key pk_2, then this leaks the vote even though it does not occur explicitly in a message. However, this cannot be detected by considering individual states. Both approaches can work with both deductive and equational models of knowledge, although the state-based approach generally uses deductive models and the equivalence-based approach generally uses equations. Both approaches also support protocol verification both with respect to active attackers (who can intercept and manipulate messages) and passive attackers (who can just intercept). In addition, using equivalences, properties can be defined with respect

[11] Blanchet (2004)

Property	State-Based	Equivalence-Based	*This Thesis*
Secrecy	yes	yes	*yes*
Unlinkability	yes†	yes	*yes*
Id-Data Privacy	yes†	yes	*yes*
Anonymity	no	yes	*yes*
Involvement	no	yes	*yes*
Knowledge model	mostly deductive	mostly equational	*both*
Attacker model	passive/active	passive/active/ receipt†/coercion†	*insider*

Table 9.8: Types of privacy property considered in the literature: rows represent properties; columns show which approach have been used to verify them. The bottom rows show with respect to what kinds of knowledge and attacker model each property can be verified with the respective approaches. †: property cannot be verified automatically

to an attacker that can try to force honest actors to provide receipts showing that they performed certain actions, or who can coerce them[12]. Intuitively, these properties are defined in terms of the existence of an equivalent process in which the honest actor provides a fake receipt, or fakes cooperation with the attacker. In particular, no tools exist that can verify (some instances of) these properties automatically, but the correctness of a manual construction of this equivalent process reduces to a normal equivalence to which the above methods can be applied.

[12] E.g., Delaune et al. (2009), Dreier et al. (2012), Dong et al. (2013)

The approach presented in this thesis is similar to the state-based approach in that we also analyse knowledge from a single system instance. In particular, we also do not take into account implicit flows. Whereas the above approaches analyse the knowledge of one (outside) attacker in the system instance, we analyse the knowledge of multiple insiders in the system. Hence, rather than "active" or "passive", we call our attacker model "insider". However, the above approaches can also be used to analyse knowledge of a particular insider (or coalition of insiders) by giving the attacker access to its secrets; and our methods can also be used to analyse passive attackers by keeping track of all messages they may observe[13]. Unlike the two above approaches, we do not analyse knowledge for all system evolutions possible with a certain attacker behaviour: instead, we consider knowledge in one regular system evolution. This is a weaker kind of analysis; but adopting it allows us to obtain practical and automated analysis results, as demonstrated in this thesis.

[13] This is indeed considered in the comparison in Section 9.3

9.2 *Privacy Properties*

In Table 9.8, we show the main types of privacy property found in the literature. Namely, we identify five types of property (secrecy, unlinkability, id-data privacy, anonymity, and involvement) that each concern a different type of knowledge. The table shows both which of the approaches from Section 9.1 can be used to verify them; and with respect to what kinds of attacker model they are defined.

Secrecy-type privacy properties state that an actor cannot determine a particular piece of personal information about a data subject. Different variants of this property exist. The traditional variant, sometimes called *weak secrecy*[14], asks if the actor can determine the contents of the information. *Resistance to guessing attacks*[15] is a

[14] Blanchet (2004)

[15] Corin et al. (2005); and Delaune et al. (2008)

stronger property: for it to hold, the actor should not be able to determine the information even if he can guess its contents. Because attributes representing personal information (e.g., street name, age) usually have relatively few possible values, this model is more realistic. *Strong secrecy* is even stronger. Intuitively, for it to hold, the actor should not be able to recognise the information even if he could influence its contents. We mention it here for completeness; for privacy analysis, this property is generally too strong. Secrecy-type privacy properties include secrecy in SAML single sign-on[16] and key establishment[17] (based on weak secrecy); and data privacy[18], paper/score/review secrecy in electronic conference management[19], and strong bidding-price secrecy of auctions[20] (based on resistance to guessing attacks). Intuitively, weak secrecy corresponds to our concept of derivability; as we argued in Section 4.2, resistance to guessing attacks corresponds to our notion of equatability if we include a correct guess in our model. We do not capture strong secrecy.

[16] Armando et al. (2008)

[17] Tounsi et al. (2012)

[18] Dong et al. (2013)

[19] Arapinis et al. (2012)

[20] Dreier et al. (2013)

Weak secrecy is naturally formalised as a state-based property. It can be verified using any tool for state-based protocol verification, e.g., AVISPA[21], ProVerif[22], or Tamarin[23]. Indeed, it has been used to formalise privacy properties[24]. Resistance to guessing attacks[25] and strong secrecy[26] have also been considered in state-based approaches, but not, as far as we know, to verify privacy properties. On the other hand, resistance to guessing attacks is naturally formalised as an equivalence-based property by considering equivalence when replacing a piece of information by a random value; also this property has been used to define and verify privacy properties[27].

[21] Armando et al. (2005)

[22] Blanchet and Smyth (2011)

[23] Schmidt et al. (2012)

[24] E.g., Armando et al. (2008), Tounsi et al. (2012)

[25] Corin et al. (2003); and Lowe (2004)

[26] Blanchet (2004)

[27] E.g., Arapinis et al. (2012), Dong et al. (2013)

Unlinkability properties state that an actor does not know that the same data subject has been involved in several protocol instances. Typically, the actor cannot observe that the same identifier occurs in messages from both protocol instances[28]. Properties of this type include strong and weak unlinkability[29]; untraceability[30]; and (strong) doctor untraceability in e-health[31]. In our model, this property corresponds to associability of the context representing the data subject in two protocol instances.

[28] Depending on the formalisation, the actor may or may not know the identifier as part of his initial knowledge

[29] Arapinis et al. (2010)

[30] van Deursen et al. (2008)

[31] Dong et al. (2012)

In the state-based setting, van Deursen et al.[32] propose a definition for unlinkability based on the concept of "reinterpreting" messages with respect to an actor. Intuitively, they define which parts of a message can be changed without the actor noticing. They define unlinkability by stating that for every system evolution in which there is a link, it should be possible to unnoticeably change the messages to messages from a system evolution in which there is no link. However, van Deursen et al.[33] do not propose a way to automatically verify this property. In the equivalence-bases setting, unlinkability properties are modelled as equivalences between an execution in which a known actor is involved in a single protocol instance, and an execution in which he is involved in two instances[34]; or as equivalence between an execution in which all actors are only involved in one protocol instance and one in which they are not[35].

[32] van Deursen et al. (2008)

[33] van Deursen et al. (2008)

[34] E.g., Arapinis et al. (2010), Dong et al. (2012)

[35] E.g., Arapinis et al. (2010), Dong et al. (2012)

Id-Data Privacy properties state that an actor cannot determine that a particular piece of personal information refers to a particular data subject. This property is particularly relevant in settings such as e-voting or e-auctions in which actors may learn both which data subjects were involved in protocol instances (e.g., which voters cast a vote) and which pieces of personal information were exchanged (e.g., which votes), but not their combination (e.g., which voter cast which vote). Properties of this type include privacy in electronic toll collection[36] and electronic voting[37]; bidding-price unlinkability and weak anonymity in e-auctions[38]; and prescription privacy in e-health[39]. Intuitively, such properties are expressed in our model by combining associability and detectability in one context.

[36] Dahl et al. (2011)

[37] Butin et al. (2013); and Delaune et al. (2009)

[38] Dreier et al. (2013)

[39] Dong et al. (2012)

Recent efforts[40] aim to capture id-data privacy in e-voting systems (e.g., the property that a vote cannot be linked to the voter) using the state-based inductive method[41]. Intuitively, the inductive method proposes to model reachable states inductively, and to use a theorem prover for showing that certain knowledge cannot be derived in any reachable state. To support id-data secrecy, Butin et al.[42] define a set of "association rules" specifying that pieces of information from the same message can be linked to each other and to the message's sender and receiver; and that "common elements" can be used to link information from different messages. However, these "common elements" and exceptions to linking rules have to be defined ad-hoc for a particular protocol. Moreover, the inductive method is inherently non-automated. In the equivalence-based setting, id-data privacy is typically defined by considering equivalence of two instances of a system with equal identifiers in which the data items are swapped. Adaptations have been proposed for specific application domains: for instance, in e-auctioning, only the winning bid should become known[43]; in e-voting, votes with different weights can be considered[44].

[40] Butin and Bella (2012); and Butin et al. (2013)

[41] Paulson (1998)

[42] Butin et al. (2013)

[43] Dreier et al. (2013)

[44] Dreier et al. (2012)

Anonymity properties state that an actor does not know that a particular data subject has been involved in a protocol instance. Typically, the actor knows the identifier of the data subject, and this property means that he does not know whether or not that identifier occurs in the protocol instance. Properties of this type found in the literature include strong and (weak) anonymity[45]; and doctor and patient anonymity in e-health[46]. In our model, the actor knows, e.g., identifier $id|_u^\kappa$ as part of his initial knowledge; anonymity for data subject ds in protocol instance $*|_{ds}^\pi$ means that he cannot associate $*|_u^\kappa$ and $*|_{ds}^\pi$.

[45] Arapinis et al. (2010)

[46] Dong et al. (2012)

In the equivalence-based setting, anonymity properties are formalised as equivalence of an execution in which a known actor participates and an execution in which he does not[47]. We are not aware of formalisations of anonymity using state-based techniques; however, such formalisations may be possible by adapting existing definitions of resistance to guessing attacks.

[47] E.g., Arapinis et al. (2010), Dong et al. (2012)

Finally, *involvement properties* state that an actor should not be able to determine that two specific parties were involved in the

same protocol instance. For instance, the actor should not be able to know that a particular data subject was involved in a protocol with a particular identity provider. As far as we are aware, the only formalisation of such a property in the literature is that of author-reviewer unlinkability in electronic conference management[48]. In our model, this property corresponds to associability of two contexts from the same domain to two separate external contexts.

[48] Arapinis et al. (2012)

The only formalisation of an involvement property[49] from the literature is equivalence-based and essentially formalises it as an id-data privacy property.[50] We are not aware of formalisations of involvement properties using state-based techniques; however, such formalisations may be possible by adapting definitions for id-data privacy as described above.

[49] Arapinis et al. (2012)

[50] Indeed, note that also in our model, these two types of property have similar formalisations.

In summary, our approach, like the equivalence-based approach, can be used to model the main privacy properties in the literature. The state-based approach is used by much fewer works, so it is not clear to what extent it can be generalised to cover additional properties. Finally, we mention that the above approaches all formalise privacy properties by modelling a particular (family of) scenario(s) that is assumed to be representative. For instance, in this thesis, we model a representative set of pieces of information, and formalise privacy properties as knowledge about this particular information. Several works[51] instead define variants of the above properties in more general, abstract frameworks, and then prove relations between these variants. However, in general, no automated techniques exist to verify properties defined in these frameworks. Also, it is not clear if the different variants of privacy properties defined in such frameworks actually reveal interesting privacy differences about communication protocols. Indeed, Brusó et al.[52] show that under reasonable conditions, several variants of privacy properties actually coincide.

[51] E.g., Bohli and Pashalidis (2011), Brusó et al. (2013a,b), Hevia and Micciancio (2008), Langer et al. (2010), Pfitzmann and Hansen (2009)

[52] Brusó et al. (2013b)

9.3 *Comparing Our Model to Equivalence-Based Properties*

In the above section, we described at a high level how different kinds of properties are formalised in different frameworks. In particular, we showed how secrecy (in the sense of resistance to guessing attacks), unlinkability, id-data privacy, and anonymity can be formalised both using our framework, and using the equivalence-based approach. Because the equivalence-based approach is the most comprehensive and actively-researched alternative to our framework, we now discuss the correspondence between our formalisations and those using equivalences in more detail.[53]

[53] We do not include involvement properties in this comparison. This is to keep the scenario we use for our comparison simple. However, as noted above, involvement properties are formalised similarly to id-data properties, so similar conclusions should apply.

In particular, we consider privacy properties in a simple scenario, which we model as a Personal Information Model. We define a scenario with three instances π_1, π_2, π_3 of a communication protocol aiming to protect the identity of a data subject with identifier $id|_{ds}^{\pi_1}, id|_{ds}^{\pi_2}, id|_{ds}^{\pi_3}$, as well as the contents of a piece of information $d|_{ds}^{\pi_1}, d|_{ds}^{\pi_2}, d|_{ds}^{\pi_3}$ about him. In this scenario, protocol instances π_1 and

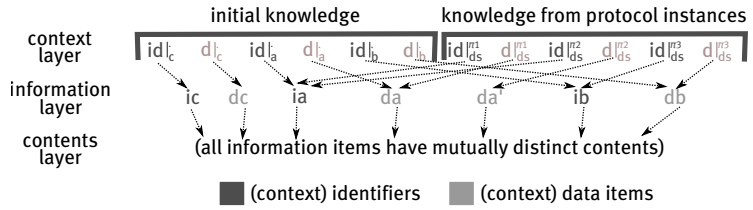


Figure 9.1: PI Model for the scenario to compare equivalence-based privacy properties with our model

Privacy Property	Initial Knowledge	Consequence in scenario
Anonymity of *ia* (I)	$\{id\|^{\cdot}_{a}\}$	$\mathcal{C} \not\vdash id\|^{\cdot}_{a} \doteq_0 id\|^{\pi_1}_{ds} \wedge \mathcal{C} \not\vdash id\|^{\cdot}_{a} \doteq_0 id\|^{\pi_2}_{ds}$
Anonymity of *ia* (II)	$\{id\|^{\cdot}_{a}, id\|^{\cdot}_{c}\}$	$\mathcal{C} \not\vdash id\|^{\cdot}_{a} \doteq_0 id\|^{\pi_1}_{ds} \wedge \mathcal{C} \not\vdash id\|^{\cdot}_{a} \doteq_0 id\|^{\pi_2}_{ds}$
Unlinkability (I)	$\varnothing$	$\mathcal{C} \not\vdash id\|^{\pi_1}_{ds} \doteq_0 id\|^{\pi_2}_{ds}$
Unlinkability of *ia* (II)	$\{id\|^{\cdot}_{a}, id\|^{\cdot}_{c}\}$	$\neg(\mathcal{C} \vdash id\|^{\cdot}_{a} \doteq_0 id\|^{\pi_1}_{ds} \wedge \mathcal{C} \vdash id\|^{\cdot}_{a} \doteq_0 id\|^{\pi_2}_{ds}) \wedge \neg(\mathcal{C} \vdash id\|^{\cdot}_{a} \doteq_0 id\|^{\pi_1}_{ds} \wedge \mathcal{C} \vdash id\|^{\pi_1}_{ds} \doteq_0 id\|^{\pi_2}_{ds}) \wedge$ $\neg(\mathcal{C} \vdash id\|^{\cdot}_{a} \doteq_0 id\|^{\pi_2}_{ds} \wedge \mathcal{C} \vdash id\|^{\pi_2}_{ds} \doteq_0 id\|^{\pi_1}_{ds})$
Secrecy of *da* (I)	$\{d\|^{\cdot}_{a}, d\|^{\cdot}_{c}\}$	$\mathcal{C} \not\vdash d\|^{\cdot}_{a} \doteq_0 d\|^{\pi_1}_{ds}$
Secrecy of *da* (II)	$\{id\|^{\cdot}_{a}, d\|^{\cdot}_{a}, d\|^{\cdot}_{c}\}$	$\mathcal{C} \not\vdash d\|^{\cdot}_{a} \doteq_0 d\|^{\pi_1}_{ds}$
Id-Data Privacy of *ia*, *da*, *ib*, *db*	$\{id\|^{\cdot}_{a}, d\|^{\cdot}_{a}, id\|^{\cdot}_{b}, d\|^{\cdot}_{b}\}$	$\neg(\mathcal{C} \vdash d\|^{\cdot}_{b} \doteq_0 d\|^{\pi_3}_{ds} \wedge \mathcal{C} \vdash id\|^{\cdot}_{b} \doteq_0 id\|^{\pi_3}_{ds}) \wedge$ $\neg(\mathcal{C} \vdash d\|^{\cdot}_{a} \doteq_0 d\|^{\pi_1}_{ds} \wedge (\mathcal{C} \vdash id\|^{\pi_1}_{ds} \doteq_0 id\|^{\cdot}_{a} \vee (\mathcal{C} \vdash id\|^{\pi_1}_{ds} \doteq_0 id\|^{\pi_2}_{ds} \wedge \mathcal{C} \vdash id\|^{\pi_2}_{ds} \doteq_0 id\|^{\cdot}_{a})))$

Table 9.9: Equivalence-based properties translated to the three-layer model: initial knowledge of the attacker and consequences of the property

π_2 have the same data subject with the same identifier; the data subject of π_3 is different. All three pieces of information are different. We also include context items $*|^{\cdot}_{*}$ representing possible initial knowledge by the attacker.[54] The Personal Information Model is shown in Figure 9.1.

To relate equivalence-based privacy properties to knowledge about this Personal Information Model, consider a passive attacker who can see which messages belong to which protocol instance. In general, equivalence-based privacy properties state that this attacker should not be able to distinguish two similar instances A, B of a system. Suppose the above scenario is an evolution of system instance A. Then there should be an evolution of system instance B that looks similar to the attacker. In particular, if the attacker can equate, e.g., $id|^{\pi_1}_{ds}$ and $id|^{\pi_2}_{ds}$ in the scenario, then there should also be an evolution of system instance B in which he can equate two identifiers. As a consequence, if there are no evolutions of system instance B with content equivalent identifiers, we can conclude that the attacker cannot equate $id|^{\pi_1}_{ds}$ and $id|^{\pi_2}_{ds}$ in the Personal Information Model.

Using the above intuition (which we make more precise below), we can translate equivalence-based privacy properties into properties about the attacker's knowledge in the above scenario. The results for several variants of the privacy properties from Section 9.2 are shown in Table 9.9. Namely, we consider two types of anonymity property: for the first one[55], the initial knowledge of the attacker consists of just $id|^{\cdot}_{a}$; for the second one[56], it also includes $id|^{\cdot}_{c}$. Both correspond to the intuitive meaning of anonymity: namely, that the attacker cannot observe that identifier $id|^{\cdot}_{a}$ occurs in one of the protocol instances. For unlinkability, we also consider two types. For the first type[57], the attacker does not have any initial knowledge and he is not able to link the two sessions with the same identifier. For the second type[58], the attacker knows identifier $id|^{\cdot}_{a}$ (as well as $id|^{\cdot}_{c}$, for technical reasons), and he should not be able to link two sessions

[54] This initial knowledge includes items $id|^{\cdot}_{c}, d|^{\cdot}_{c}$ that do not occur in the protocol instances. This is for technical reasons; these items are not actually relevant for the attacker's knowledge

[55] Corresponding to strong anonymity in Arapinis et al. (2010) and strong doctor anonymity in Dong et al. (2012)

[56] Corresponding to doctor anonymity in Dong et al. (2012)

[57] Corresponding to strong unlinkability in Arapinis et al. (2010) and strong doctor untraceability in Dong et al. (2012)

[58] Corresponding to doctor untraceability in Dong et al. (2012)

to each other *of which he knows the data subject*. There are also two types of secrecy. Both express that the actor should not be able to see that the given data item was used in the protocol instance. Their difference is technical: for the first type[59], the actor does not know the identifier $id|^{\cdot}_{a}$ of the relevant data subject; for the second type[60], he does (although he does not need to be able to link it to the protocol instance). Finally, id-data privacy[61] means that neither in protocol instance π_1 nor in protocol instance π_3, the attacker should be able to determine both the identifier and the data item.[62] We conclude that the translation from equivalence-based privacy properties to our model respects the intuitive meaning of the properties.

We now show in more detail how we have translated equivalence-based privacy properties to our model. Equivalence-based privacy properties are defined as equivalences between two system instances A, B that differ in what protocol instances can occur. For instance, for anonymity, evolutions of A may include protocol instances involving an actor with a known identifier ia, evolutions of B may not. We describe equivalence-based privacy properties by modelling the sets of possible traces. Let I, D be sets of identifiers and data items, respectively. Consider system evolutions consisting of three protocol instances, each involving one identifier and one data item. Such a system evolution is captured by a *trace*: a 6-tuple $(i_1, d_1, i_2, d_2, i_3, d_3) \in (I \times D)^3$. An equivalence property prescribes sets $\mathcal{T}_1, \mathcal{T}_2$ of traces of the two system instances, and a set $\mathcal{K}$ of information known initially by the attacker. For instance, for anonymity, consider an attacker with initial knowledge $\mathcal{K} = \{ia\}$. In system instance A, each protocol instance may use any identifier (including ia) and data item, corresponding to the following traces:[63]

$$\mathcal{T}_1 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I, d_1, d_2, d_3 \in D \text{ distinct}\}.$$

In system instance B, ia may not occur, so we have traces:

$$\mathcal{T}_2 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I \setminus \{ia\}, d_1, d_2, d_3 \in D \text{ distinct}\}.$$

Hence, anonymity intuitively means that, for an attacker with initial knowledge $\mathcal{K}$, for every trace in $\mathcal{T}_1$ there exists an identically-looking trace in $\mathcal{T}_2$, and vice versa.

We now translate the anonymity property to our model. For this, we consider one particular trace $(ia, da, ia, da', ib, db) \in \mathcal{T}_1$ for system instance A. The information exchanged in this traces is modelled by the Personal Information Model I shown in Figure 9.2(a). At the context layer, we capture three protocol instances π_i, $i \in \{1, 2, 3\}$, involving identifier $id|^{\pi_i}_{ds}$ and data item $d|^{\pi_i}_{ds}$; $id|^{\cdot}_{a}$ represents initial knowledge $ia \in \mathcal{K}$. Suppose the knowledge of the attacker from the above trace is represented by a knowledge base $\mathcal{C} = \{id|^{\cdot}_{a}, \ldots\}$ with respect to the above Personal Information Model[64]. For anonymity to hold, there should be a trace in $\mathcal{T}_2$ with identically-looking knowledge. For instance, consider trace $(ib, da, ib, da', ib, db) \in \mathcal{T}_2$. The information in this trace can be represented by an Information Model I' similar to that in Figure 9.2(a), but where $id|^{\pi_1}_{ds}$ and $id|^{\pi_2}_{ds}$

[59] Corresponds to paper, score, and review secrecy in Arapinis et al. (2012)

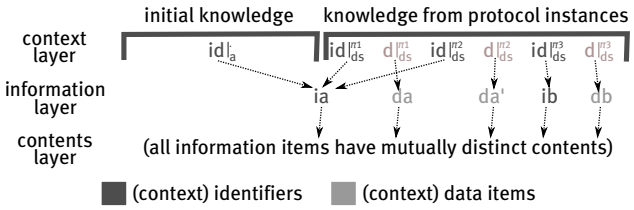
[60] Corresponding to data-privacy in Dong et al. (2013) and strong bidding-price secrecy of auctions in Dreier et al. (2013)

[61] Corresponding to privacy in electronic toll collection in Dahl et al. (2011) and electronic voting in Delaune et al. (2009); bidding-price unlinkability and weak anonymity in e-auctions in Dreier et al. (2013); and prescription privacy in e-health in Dong et al. (2012)

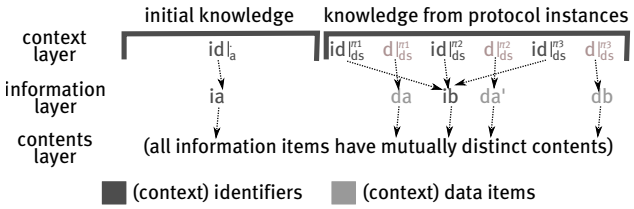
[62] Note that both direct equatability $id|^{\pi_1}_{ds} \doteq_0 id|^{\cdot}_{a}$ and indirect equatability $id|^{\pi_1}_{ds} \doteq_0 id|^{\pi_2}_{ds} \wedge id|^{\pi_2}_{ds} \doteq_0 id|^{\cdot}_{a}$ are considered.

[63] Note that the data items in the different protocol instances are distinct, but the identifiers could be the same

[64] The Personal Information Model may include additional pieces of information



(a) Information model for anonymity scenario in system instance A



(b) Information model for a possibly identically scenario in system instance B

Figure 9.2: Personal Information Models for the anonymity property

map to *ib* instead of *ia* (Figure 9.2(b)). The attacker's knowledge from executing trace $\mathcal{T}_2$ can be described by the same knowledge base $\mathcal{C}$, but with respect to the different Information Model I'[65]. As argued before, if $\mathcal{C}$ looks identical in I and I', then context items that are not content equivalent in I', cannot be equatable in I. (If they were equatable in I, then they should also be equatable in I', and in particular, they should be content equivalent.)[66] In this case, this means that $\mathcal{C} \not\vdash id|_a^{\cdot} \doteq_0 id|_{ds}^{\pi_1}$ and $\mathcal{C} \not\vdash id|_a^{\cdot} \doteq_0 id|_{ds}^{\pi_2}$. By similarly analysing all other traces in $\mathcal{T}_2$, one finds that all of them imply $\mathcal{C} \not\vdash id|_a^{\cdot} \doteq_0 id|_{ds}^{\pi_1} \wedge \mathcal{C} \not\vdash id|_a^{\cdot} \doteq_0 id|_{ds}^{\pi_2}$. Hence, intuitively, if anonymity holds, then $\mathcal{C} \not\vdash id|_a^{\cdot} \doteq_0 id|_{ds}^{\pi_1} \wedge \mathcal{C} \not\vdash id|_a^{\cdot} \doteq_0 id|_{ds}^{\pi_2}$ needs to be true in our model.

The same reasoning can be applied to translate the other properties to our model. For this, we need to model each property as a set $\mathcal{K}$ of initial knowledge, and two sets $\mathcal{T}_1, \mathcal{T}_2$ of possible traces (Figure 9.3). We now discuss the models in the figure. Above, we discussed variant I of anonymity[67]. Another variant in the literature[68] allows a known identifier *ic* to occur in traces in $\mathcal{T}_2$. There are also two variants of unlinkability: variant I[69] compares traces in which all identifiers are distinct to traces in which they are not; variant II[70] compares traces in which known identifier *ia* occurs at most twice to traces in which known identifiers *ia*, *ic* both occur at most once. Similarly, two variants of secrecy consider secrecy of a data item in combination with an unknown identifier (variant I[71]) or a known identifier (variant II[72]). For id-data privacy[73], *da* occurs with *ia* and *db* with *ib* in one set of traces, and *da* occurs with *ib* and *db* with *ia* in the other. Note that, for each model in Figure 9.3, $(ia, da, ia, da', ib, db) \in \mathcal{T}_1$[74]. Hence, we can translate all privacy properties to knowledge about a single Personal Information Model I'' obtained from I (Figure 9.2(a)) by adding context items for each piece ib, ic, da, db, dc of possible initial knowledge. Indeed, this gives the Personal Information Model shown in Figure 9.2 and translation results shown in Table 9.9.

[65] This assumes that the identifiers and data items exchanged in the protocol instances do not influence other parts of the message, i.e., we assume there are no implicit flows.

[66] Indeed, note that Lemma 4.5.4 formalises exactly this intuition in the setting of equational theories corresponding to rule-based models.

[67] Corresponding to strong anonymity in Arapinis et al. (2010) and strong doctor anonymity in Dong et al. (2012)

[68] Corresponding to doctor anonymity in Dong et al. (2012)

[69] Corresponding to strong unlinkability in Arapinis et al. (2010) and strong doctor untraceability in Dong et al. (2012)

[70] Corresponding to doctor untraceability in Dong et al. (2012)

[71] Corresponds to paper, score, and review secrecy in Arapinis et al. (2012)

[72] Corresponding to data-privacy in Dong et al. (2013) and strong bidding-price secrecy of auctions in Dreier et al. (2013)

[73] Corresponding to privacy in electronic toll collection in Dahl et al. (2011) and electronic voting in Delaune et al. (2009); bidding-price unlinkability and weak anonymity in e-auctions in Dreier et al. (2013); and prescription privacy in e-health in Dong et al. (2012)

[74] and $(ia, da, ia, da', ib, db) \notin \mathcal{T}_2$

Anonymity (I) : $\mathcal{K} = \{ia\}$

$\mathcal{T}_1 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I, d_1, d_2, d_3 \in D \text{ distinct}\}$

$\mathcal{T}_2 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I \setminus \{ia\}, d_1, d_2, d_3 \in D \text{ distinct}\}$

Anonymity (II) : $\mathcal{K} = \{ia, ic\}$

$\mathcal{T}_1 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \in \mathcal{T} \mid i_1, i_2, i_3 \in I \setminus \{ic\}, d_1, d_2, d_3 \in D \text{ distinct}\}$

$\mathcal{T}_2 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \in \mathcal{T} \mid i_1, i_2, i_3 \in I \setminus \{ia\}, d_1, d_2, d_3 \in D \text{ distinct}\}$

Unlinkability (I) : $\mathcal{K} = \varnothing$

$\mathcal{T}_1 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I, d_1, d_2, d_3 \in D \text{ distinct}\}$

$\mathcal{T}_2 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I \text{ distinct}, d_1, d_2, d_3 \in D \text{ distinct}\}$

Unlinkability (II) : $\mathcal{K} = \{ia, ic\}$

$\mathcal{T}_1 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I \setminus \{ic\}, \#\{n | i_n = ia\} \leq 2, d_1, d_2, d_3 \in D \text{ distinct}\}$

$\mathcal{T}_2 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I, \#\{n | i_n = ia\} \leq 1, \#\{n | i_n = ic\} \leq 1, d_1, d_2, d_3 \in D \text{ distinct}\}$

Secrecy (I) : $\mathcal{K} = \{da, dc\}$

$\mathcal{T}_1 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in J, d_1, d_2, d_3 \in D \setminus \{dc\} \text{ distinct}\}$

$\mathcal{T}_2 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in J, d_1, d_2, d_3 \in D \setminus \{da\} \text{ distinct}\}$

Secrecy (II) : $\mathcal{K} = \{ia, da, dc\}$

$\mathcal{T}_1 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I, d_1, d_2, d_3 \in D \setminus \{dc\} \text{ distinct}, d_k = da \Rightarrow i_k = ia\}$

$\mathcal{T}_2 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I, d_1, d_2, d_3 \in D \setminus \{da\} \text{ distinct}, d_k = dc \Rightarrow i_k = ia\}$

Id-Data Privacy : $\mathcal{K} = \{ia, ib, da, db\}$

$\mathcal{T}_1 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I, d_1, d_2, d_3 \in D \text{ distinct}, d_k{=}da \Rightarrow i_k{=}ia, d_k{=}db \Rightarrow i_k{=}ib\}$

$\mathcal{T}_2 := \{(i_1, d_1, i_2, d_2, i_3, d_3) \mid i_1, i_2, i_3 \in I, d_1, d_2, d_3 \in D \text{ distinct}, d_k{=}db \Rightarrow i_k{=}ia, d_k{=}da \Rightarrow i_k{=}ib\}$

Figure 9.3: Sets of traces corresponding to privacy properties

9.4 Discussion

The privacy guarantees offered by protocols can also be analysed without verifying privacy properties. In the computational model of cryptography (of which the formal models from the above approaches are an abstraction), privacy can be captured implicitly by modelling the *ideal functionality*[75] of protocols. Namely, this ideal functionality specifies what information each actor in a protocol should learn. If a protocol is secure with respect to this ideal functionality, then in particular, no actor in the protocol can learn any additional information. Based on this idea, privacy in anonymous credentials schemes[76], smart metering[77], and watermarking[78] has been proven. However, note that these proofs are all manual, and operate at a much lower level of abstraction than the above methods.

[75] Beaver (1991)

[76] Camenisch and Lysyanskaya (2001)

[77] Rial and Danezis (2011)

[78] Rial et al. (2011)

Based on formal methods approaches, Mödersheim et al.[79] propose a way to define privacy in terms of intended disclosure, and to reason about what equivalences this privacy corresponds to. Although this seems a promising approach, there is currently no way to integrate it with automated verification techniques.

[79] Mödersheim et al. (2013)

10
Conclusions

Contents

WE STARTED THIS THESIS by introducing two different proposals for a system to distribute patient data to researchers, and asking: "From the point of view of patient privacy, which proposal would you pick?". Generalising this example, we obtained the following research question:

How can we rigorously understand the privacy impact of information exchange in distributed systems?

In this thesis, we have presented techniques that provide an answer to this question. Specifically, we have focussed on how the use of particular communication protocols for information exchange influences what privacy-sensitive information is revealed. In particular, we have considered the knowledge of the actors in the system, i.e., insiders who are legitimately involved in the day-to-day operation of the system. Also, weve interpreted privacy as "data minimisation", i.e., making sure that actor can derive as little information as possible (not considering consent, user experience, or other privacy issues). We have demonstrated the usefulness of these techniques by, among other things, showing how they can be used to gain insight into the privacy merits of the two proposed systems for distributing patient data to researchers discussed above. Namely, as we have rigorously shown in Chapter 8, one proposal requires actors to store more data than is desirable; the other proposal does not, but only improves privacy with respect to non-curious actors (we have also suggested possible improvements).

In Chapter 1, we have argued that, to answer our research question, we need techniques that satisfy three basic requirements: to make our analysis rigorous, they need to provide precise and verifiable results (requirement 1); to make our analysis useful, these results need to be easy to interpret (requirement 2); yet to make

analysis feasible in practice, they should be largely automated (requirement 3).

We now discuss our answer to the research question (§10.1); the limitations of our proposed techniques (§10.2); and some possible directions for future work (§10.3).

10.1 Contributions

A first step towards answering the research question is to have a model in which we can define precise but general "privacy properties" that express relevant privacy concerns. This motivated our first sub-question:

Question 1. How can we represent privacy properties about actors in distributed systems in a system-independent way?

To answer this question, in Chapter 2 we have proposed the *Personal Information Model*, a model in which the knowledge of actors about personal information can be precisely expressed. This model is high-level, but can nonetheless be used to precisely capture a range of privacy properties; in particular, any property that can be translated to fundamental (un)detectability, (un)linkability and (non-)involvement properties (Section 2.3). In Chapter 7, we showed that this model is expressive enough to capture a comprehensive set of privacy concerns in the context of identity management (requirement 1). We also proposed a visualisation of the knowledge of all actors and coalitions in a system, by means of coalition graphs (Section 2.4), by which our model can be easily interpreted (requirement 2). In Chapter 5, we generalised the Personal Information Model to the Symbolic Information Model, making it independent from characteristics of a particular scenario (e.g., the number of parties involved and the amount of personal information exchanged).

Compared to the literature, both our high-level model for privacy properties and its visualisation are new. In existing work[1], privacy properties are generally represented on an ad-hoc basis depending on the particular protocol (making it difficult to compare different systems); those general encodings that exist generally do not consider information about third parties that occurs in applications like identity management (see Chapter 7) and patient data pseudonymisation (see Chapter 8). Also, we are not aware of existing ways to visualise the knowledge of all actors in an information system that has a precise interpretation in an underlying model like our coalition graphs.

[1] E.g., Arapinis et al. (2012), Dahl et al. (2011), Dong et al. (2013)

Having shown how to represent privacy properties, the next step is to decide if they hold for a particular system. To analyse knowledge of actors, the messages they have exchanged are commonly modelled using formal methods techniques, in which the cryptographic primitives used for communication are modelled as abstract "black boxes". In this thesis, we decided to adopt such techniques, hence we need to decide which properties hold based on a formal

representation of messages. This has lead to our second research question:

Question 2. How can we automatically decide privacy properties based on a formal model of information exchange?

We have provided three alternative answers to this question, relying on three slightly different formal representations of messages. The first method (Chapter 3) generalises existing approaches that model cryptographic primitives using inference rules. For this method, we present a terminating algorithm to decide privacy properties from a model of messages, along with an implementation. The second method (Chapter 4) uses existing models of cryptographic primitives using equational theories. This model allows a wider range of cryptographic primitives to be (more accurately) modelled. We show how privacy properties can be decided with the help of existing tools for message analysis using equational theories. Unfortunately, the performance of this method is much worse than that of the first method. The third method (Chapter 5) is a generalisation of the first method to achieve scenario-independent results using the Symbolic Information Model. Again, we present a terminating algorithm and an implementation (requirement 3).

Compared to the literature, our proposed methods rely on existing ideas but apply them in new ways. Previous works have also analysed knowledge of actors using inference rules[2]. We made significant adaptations to these works to reason not just about what pieces of information an actor knows, but also what message or protocol instance he leans them from. Other works[3] have also analysed knowledge of actors using equational theories, in particular expressing that an actor knows nothing about a copy of a piece of information. We show how equational theories can be used to dually express that an actor can determine exactly what contents a copy of a piece of information has. The methods in Chapter 5 bears some similarity to existing protocol analysis using constraint systems[4], but have been developed independently. Finally, we contribute novel models for zero-knowledge proofs (Sections 6.4) and anonymous credentials (Sections 6.5).

[2] E.g., Clarke et al. (1998), Boreale (2001)

[3] E.g., Blanchet (2004), Corin et al. (2005)

[4] E.g., Comon-Lundh et al. (2010)

Having proposed formal techniques for modelling and deciding privacy properties, we finally need to show how these techniques are used for our higher-level goal, i.e., to obtain an understanding of the privacy impact of information exchange. More concretely, we need to show that, starting from a number of systems we want to analyse, the intended understanding can be obtained by performing a number of well-defined steps. This has lead to our third and final research question:

Question 3. Which steps need to be followed to actually analyse privacy impact of information exchange?

We have answered this question by performing two concrete case studies that demonstrate two ways in which our techniques can be

used to understand privacy impact. Namely, one way is to verify a given set of properties; another way is to visually comparing privacy in different systems using constraint graphs. In both cases, we show in general terms what steps need to be followed, and then demonstrate these steps in a concrete case study. These steps use our tools[5] for the formal analysis of privacy in communication protocols based on the above techniques. In Chapter 7, we have contributed a comprehensive set of privacy requirements for identity management, as well as formal models of four different identity management systems. In Chapter 8, we have contributed a rigorous analysis of achievable privacy guarantees for patient data pseudonymisation.

[5] The tools and the formal models from the case studies are available at `http://code.google.com/p/objective-privacy/`

Compared to the literature, we believe that we are the first to provide the complete machinery to perform a formal privacy comparison of different systems. First, we allow protocol-independent specification of properties that relate personal information to its data subject, whereas existing protocol-independent frameworks consider knowledge of links to its sender[6] or knowledge of the personal information itself[7]. Second, we propose terminating algorithms for property verification, whereas the technical complexity of existing (e.g., equivalence-based) definitions makes automated verification difficult. Indeed, existing analyses of systems involving more than a few only consider privacy properties about some of these actors[8] or consider systems that have been designed with automated verification in mind[9].

[6] Arapinis et al. (2010)

[7] Dong et al. (2013)

[8] E.g., Dong et al. (2012)

[9] Arapinis et al. (2012)

In conclusion, the tools developed in this thesis have given us new ways to understand how the use of different communication protocols influences what privacy-sensitive information the actors in a system learn. This constitutes our answer to the research question posed above.

10.2 Limitations of the Proposed Techniques

As argued above, our techniques provide an answer to the above research question; but by focussing on particular aspects of the question, we leave other aspects unconsidered. We now discuss several of them.

Interpretation of information First, by focussing on communication protocols, we do not take into account privacy impact due to interpretation of the information exchanged. For instance, we do not consider how combinations of exchanged attributes like address, city of birth, and age might be used to identify people. Two different research streams consider this kind of analysis: one research stream considers how to experimentally link given data; the other considers how to guarantee that such linking is impossible. Methods to link data using non-identifiers have been investigated since the seminal paper by Fellegi and Sunter[10]; Köpcke and Rahm[11] provide a recent comparison of available systems. Data from more than two sources can be grouped together based on pairwise decisions using domain-

[10] Fellegi and Sunter (1969)

[11] Köpcke and Rahm (2010)

dependent[12] or domain-independent[13] algorithms, or statistical techniques[14]. On the other hand, statistical methods to guarantee that exchanged personal information cannot be linked to other data include k-anonymity, ℓ-diversity, t-closeness and differential privacy[15]. Koot[16] reports on experiments in which the "anonymity" of particular disclosures is quantified. Inferring the values of some attributes from others is covered, e.g., by Pontes et al.[17]. Other recent work[18] analyses privacy-friendly release of data with interdependencies.

[12] E.g., Bhattacharya and Getoor (2007), Parag and Domingos (2004), Mray et al. (2007), Sapena et al. (2008)

[13] E.g., Bilenko et al. (2005), Chaudhuri et al. (2005)

[14] E.g., Sadinle and Fienberg (2012)

[15] k-anonymity: Samarati and Sweeney (1998), Samarati (2001); ℓ-diversity: Machanavajjhala et al. (2007); t-closeness: Li et al. (2007); differential privacy: Dwork (2006). See di Vimercati et al. (2012) for an overview.

[16] Koot (2012)

[17] Pontes et al. (2012)

[18] di Vimercati et al. (2010); and di Vimercati et al. (To appear)

Need-to-know Also, we focus on analysing what information is exchanged, not on what information should be exchanged. Namely, we assume that privacy properties (capturing what information should be exchanged) and models of communication (capturing what information is exchanged) are given, and compare the two. When considering privacy in a system, it is also relevant to determine which are the relevant privacy properties. This includes determining what information exchange is really needed for the actors in the system to perform their tasks; and whether the system ensures that this information exchange is consented to by its data subject. For this, other methods can be used (cf. Compagna et al.[19]).

[19] Compagna et al. (2009)

Communication networks In addition, we focus on communication protocols but not on the underlying network used to exchange messages. Normally, communication over networks like the Internet may be traced because each packet of information contains the communication identifiers of the sender and recipient. Regardless of whether the information inside these packets can be read (which is covered by our techniques), the mere fact that communication between two parties takes place may be privacy-sensitive: for instance, consider communication with a certain hospital or bank. Various systems have been proposed to make it difficult to track messages, intuitively, by sending it via multiple random participants of the system. Chaum (1981) first introduced the concept of mix networks, and many proposals have been made since, most notably Tor[20]. Many frameworks exist that analyse privacy characteristics of such systems. Relevant privacy properties include the linkability of a message to its sender/recipient, or senders to recipients; variants of these properties are formulated informally[21], based on probabilities[22], based on which situations an attacker can distinguish[23], or using other formalisms[24].

[20] Dingledine et al. (2004)

[21] Pfitzmann and Hansen (2009)

[22] Chatzikokolakis (2007); and Steinbrecher and Köpsell (2003)

[23] Bohli and Pashalidis (2011); and Hevia and Micciancio (2008)

[24] Hughes and Shmatikov (2004); Schneider and Sidiropoulos (1996); and Syverson and Stubblebine (1999)

Other limitations Our choice to analyse privacy by analysing messages in a formal model implies some limitations. First of all, we analyse messages as specified in communication protocols. That is, we do not consider attackers (neither outside attackers that try to intercept and manipulate traffic, nor inside attackers that try to gain more knowledge than they should by manipulating messages from regular protocol instances). Also, we do not consider "implicit

flow" of information[25]. For instance, suppose that insurance company A sends a query for user attributes to actor B only if that user has had treatment X. Then even if the query itself does not include the information about the treatment, B can still derive it from the fact that the query has taken place. Existing approaches based on observational equivalences[26] do take such concerns into account, but this leads to the undecideability and over-approximation issues discussed in Chapter 1. In any case, it is important to realise that formal models like ours are high-level models that assume "perfect" underlying cryptography. For instance, it does not consider knowledge obtainable from cryptanalysis or due to faulty implementations of cryptographic primitives used in practice.

[25] Blanchet (2004)

[26] Blanchet et al. (2008)

Finally, this thesis focuses more on developing techniques for privacy analysis than on making them widely accessible. Although we have fully described and demonstrated the steps needed to perform an analysis, being able to perform these steps still requires considerable technical knowledge. In particular, our tools are all command-line and based on Prolog syntax; moreover, to be able to use them, a detailed understanding of the methods presented in this thesis is needed. While we are not aware of similar tools that provide a better user experience, this is still a limitation.

10.3 Directions for Future Work

In this final section, we discuss several possible extensions to our techniques, by which some of the above limitations may be overcome.

One interesting direction for future work is to assess the level of anonymisation of exchanged information more generally by also interpreting information. For instance, in our case study on pseudonymisation of patient data (Chapter 8), we considered how to prevent de-anonymisation based on the identifiers used when exchanging the data. However, also the patient data itself could be used for de-anonymisation: e.g., a combination of some attributes like length, eye colour, blood group, etcetera may be enough to de-anonymise a record with reasonable certainty. Hence, to obtain a full view of the privacy consequences of information exchange, we need to consider not only the knowledge gained from exchanging information, but also the knowledge gained from interpreting it after the exchange has taken place. This motivates the following research challenge:

Challenge 1. How can privacy concerns due to exchange and interpretation of information be understood in a combined fashion?

This challenge may be interpreted both on the macro scale (i.e., given many exchanges of information, what percentage of these allows recombining), or on the micro scale (i.e., in a given scenario, what particular links can be established with what degree of certainty). In both cases, other techniques are needed in addition to

formal reasoning on messages. On the macro scale, techniques for establishing pairwise links in large datasets are relevant[27], including our own experimental work[28]. On the micro scale, relevant existing techniques[29] try to determine privacy-sensitiveness of particular attribute values in a disclosure.

A second important direction for further research relates to the fact that our approach does not consider attackers. Specifically, we analyse messages according to a protocol specification, but do not consider what happens when active attackers (either insiders or outsiders) try to manipulate them. If attackers have the possibility to obtain sensitive information by manipulating messages without anybody noticing, then this clearly impacts privacy. This leads us to pose the following research challenge:

Challenge 2. How can we ensure that privacy in information exchange is not impacted by attackers?

As mentioned in Chapter 1, existing equivalence-based privacy properties that consider attackers are not general enough for a comprehensive privacy analysis, and are too difficult to verify automatically. These properties could possibly be generated from a more general model, but this would not solve the problem of automated verification. On the other hand, our tools are general enough and can be used for automated verification, but do not consider attackers. Hence, the challenge is to find a solution that combines the advantages of both approaches.

Finally, it is relevant to address the issue of usability. People for whom analysing the privacy impact of information exchange are relevant, include system designers who want to use privacy-friendly communication protocols, or system architects who want to decide what system to use. However, for such people, existing tools for privacy analysis are too hard to use and require too much background knowledge. For instance, the tools developed in the context of this thesis are command-line and based on Prolog syntax, and require a detailed understanding of the techniques developed in this thesis. Therefore, we pose the following research challenge:

Challenge 3. How can we make techniques for privacy analysis available to non-experts on communication protocols?

Clearly, for some of the tasks involved in analysing privacy, such as modelling cryptographic primitives, a deep understanding of the underlying techniques is unavoidable. However, for many other tasks, such as specifying scenarios, it should be possible to provide an intuitive graphical user interface. In Veeningen et al. (2013a), we made a first proposal for a graphical user interface for the techniques in this thesis. However, entering all information for an analysis using the proposed GUI still requires a significant amount of work. Possible ways of making the specification of scenarios more user-friendly would be to automatically generate basic scenarios that

[27] E.g., Bhattacharya and Getoor (2007), Parag and Domingos (2004), Mray et al. (2007), Sapena et al. (2008), Bilenko et al. (2005), Chaudhuri et al. (2005), Sadinle and Fienberg (2012)

[28] Veeningen et al. (2014)

[29] Koot (2012); and Ferro et al. (2013)

can then be customised by the user; and to set up "repositories" of protocols and primitives for easy re-use.

Appendix A
Samenvatting (Dutch summary)

Formeel redeneren over privacy in communicatieprotocollen

Het internet is een gevaarlijke plek. Gegevens die je via het internet uitwisselt, reizen via telkens wisselende routes van meerdere computers, die misschien niet allemaal te vertrouwen zijn. Om er toch voor te kunnen zorgen dat een ontvangen bericht niet door een crimineel[1] is vervalst, afgeluisterd, of gemanipuleerd, is een breed scala technieken bedacht onder de noemer cryptografie (oud-Grieks voor geheimschrift). Voorbeelden zijn encryptie (versleuteling), digitale handtekeningen, en cryptografische hash-functies.

Het ontwerpen van goede cryptografische basistechnieken is één ding; maar het goed gebruiken van die technieken blijkt in de praktijk iets anders. Zeker niet-experts maken vaak fouten als het gaat om subtiliteiten in de toepassing van cryptografische basistechnieken – de kopieerbeveiliging van de Sony Playstation 3 kon bijvoorbeeld gekraakt worden door een elementaire fout in het gebruik van digitale handtekeningen. Maar zelfs voor experts is het ingewikkeld om alle manieren na te gaan waarop een systeem aangevallen zou kunnen worden.

[1] Of "Nation State Adversary" (NSA)

Formeel redeneren over cryptografie

Vandaar dat, al in de jaren '80, het idee opkwam om met formele redeneertechnieken naar aanvallen op zulke systemen te gaan zoeken. Formeel redeneren betekent dat je een aantal feiten aan de computer geeft, en een aantal regels om hier nieuwe feiten uit af te leiden. In dit geval zijn de feiten de berichten (in termen van de gebruikte cryptografische technieken) die een aanvaller op een cryptografisch systeem ziet. De redeneerstappen zijn de manieren waarop een aanvaller informatie uit onderschepte berichten kan halen, en waarop hij deze berichten kan manipuleren.[2] De computer bepaalt dan of een bepaalde conclusie met de gegeven redeneerstappen uit de gegeven feiten af te leiden is; bijvoorbeeld de conclusie "de aanvaller kan het geheim achterhalen". Als dit zo is, dan heeft de computer een aanval gevonden; anders hebben we reden om aan te nemen dat het systeem veilig is. De uitdaging is nu om redeneerstappen op te stellen die veelzijdig genoeg zijn om interessante aanvallen te

[2] Hierbij nemen we aan dat de onderliggende cryptografie correct werkt. We zoeken dus naar logische fouten in het gebruik van, bijvoorbeeld, encryptie, en niet naar backdoors die door een eventuele NSA (Nation State Adversary) in de encryptie zelf zijn aangebracht.

$$\dfrac{\dfrac{}{\mathcal{C} \vdash \mathsf{enc}(\mathsf{geheim},\mathsf{sleutel})}\,(\vdash 1) \quad \dfrac{}{\mathcal{C} \vdash \mathsf{sleutel}}\,(\vdash 1)}{\mathcal{C} \vdash \mathsf{geheim}}\,(\vdash 2)$$

Figuur 1: Formele afleiding van $\mathcal{C} \vdash \mathsf{geheim}$

vinden, maar eenvoudig genoeg om door een computer efficient te berekenen.

In deze scriptie richten we ons op de vraag welke informatie een aanvaller uit berichten haalt, en niet op welke aanpassingen hij kan doen. We geven een simpel voorbeeldje om een idee te geven hoe dit soort modellen en redeneringen er uit zien: we gaan redeneren over berichten met symmetrische encryptie. Symmetrische encryptie is een cryptografische techniek om een bericht m te versleutelen met behulp van een sleutel k. Dit wordt op zo'n manier gedaan, dat alléén iemand die zelf ook sleutel k kent, de encryptie weer kan ontsleutelen om hier m uit af te leiden.[3] Laten we de encryptie van bericht m met sleutel k schrijven als $\mathsf{enc}(m, k)$ (waarbij m en k zelf ook berichten kunnen zijn, bijvoorbeeld $\mathsf{enc}(\mathsf{enc}(a, b), l)$). We willen een set redeneerstappen opstellen die vertelt welke informatie een aanvaller kent als hij de lijst $\mathcal{C}$ van berichten heeft gezien. In dit geval hebben we er twee nodig:

[3] Dit in tegenstelling tot asymmetrische encryptie, waarbij versleutelen en ontsleutelen met twee verschillende sleutels gebeurt.

Als X in de lijst $\mathcal{C}$ voorkomt, dan kent de aanvaller X. $(\vdash 1)$

Als de aanvaller $\mathsf{enc}(X, Y)$ en Y kent, dan kent hij ook X. $(\vdash 2)$

Een voorbeeld van een redenering met de bovenstaande twee regels. Een aanvaller heeft de lijst $\mathcal{C} = \{\mathsf{enc}(\mathsf{geheim}, \mathsf{sleutel}), \mathsf{sleutel})\}$ van berichten gezien: een encryptie van geheim met sleutel sleutel, en sleutel sleutel zelf. We kunnen nu met de bovenstaande redeneerregels de (nogal voor de hand liggende) conclusie afleiden dat de aanvaller het bericht geheim kent. Dit gaat zo: door regel $(\vdash 1)$ toe te passen, concluderen we dat de aanvaller $\mathsf{enc}(\mathsf{geheim}, \mathsf{sleutel})$ kent. Nogmaals regel $(\vdash 1)$ geeft dat de aanvaller óók sleutel kent. Nu kunnen we regel $(\vdash 2)$ toepassen, waarbij we voor X de encryptie en voor Y de sleutel "invullen". Dit geeft als conclusie dat de aanvaller inderdaad geheim kent.

Laten we de uitspraak "Als een aanvaller de lijst $\mathcal{C}$ van berichten heeft gezien, dan kent hij bericht X" noteren als $\mathcal{C} \vdash X$. De bovenstaande regels $(\vdash 1)$ en $(\vdash 2)$ vertellen nu voor welke $\mathcal{C}$ en X de uitspraak $\mathcal{C} \vdash X$ geldt. Met deze notatie kunnen we de bovenstaande redenering schematisch weergeven zoals in Figuur 1. Een horizontale streep staat hier voor de toepassing van een regel, waarbij de voorwaarden boven de streep worden gebruikt om de conclusie onder de streep af te leiden.

We kunnen het bovenstaande model uitbreiden door regels toe te voegen voor andere cryptografische technieken, zoals digitale handtekeningen en hash-functies. In de praktijk passen we het model natuurlijk ook toe op langere lijsten met meer ingewikkelde

berichten. Dit maakt automatisering lastig.[4]

[4] Anderzijds blijft het model een sterke versimpeling van de werkelijkheid, en er is maar weinig bekend over hoe dit model samenhangt met gedetailleerdere, realistischere modellen van de kennis van aanvallers.

Privacy: redeneren over betekenis én waarde

We hebben hierboven een idee gegeven over hoe je formeel kunt redeneren over de uitspraak "de aanvaller kent X". Op het eerste gezicht kun je dit gebruiken om een soort privacy-analyse van verschillende computersystemen te doen. Stel dat er verschillende systeemontwerpen zijn voor, bijvoorbeeld, een elektronisch patiëntendossier. We modelleren voor elk systeemontwerp de berichten die de betrokken partijen te zien krijgen. Vervolgens bekijken we welke (combinaties van samenwerkende) partijen welke informatie kunnen achterhalen: hoe minder, hoe beter.

Jammer genoeg kunnen we bestaande redeneersystemen niet direct voor zo'n analyse toepassen. Bestaande redeneersystemen maken namelijk geen onderscheid tussen informatie en haar waarde, terwijl verschillende stukjes informatie (bijvoorbeeld iemands leeftijd, of iemands huisnummer) wel dezelfde waarde kunnen hebben (bijvoorbeeld "18"). We kunnen met bestaande redeneersystemen dus informatie óf modelleren in termen van de waarde ("18"), óf in termen van de betekenis ("leeftijd van A", "huisnummer van X"), maar niet allebei. In het eerste geval kunnen we alleen afleiden of de aanvaller de waarde "18" kent, maar komen we er niet achter welke informatie met die waarde het was. Dit is voor privacy-analyse niet afdoende.

In het tweede geval echter, krijgen we óók een incompleet beeld. We moeten de stukjes informatie dan namelijk als verschillend beschouwen, waardoor redeneringen niet meer gebruik kunnen maken van het feit dat hun waarde hetzelfde is. Stel, ik leer een encryptiesleutel als zijnde "de sleutel van Meilof", en zie later een bericht versleuteld met "de sleutel van Geert", die desondanks dezelfde waarde heeft. Regel ($\vdash$2) hierboven zegt dat we de sleutel van Meilof niet kunnen gebruiken om het versleutelde bericht te ontcijferen. We onderschatten hiermee de informatie die een aanvaller kan achterhalen, wat voor privacy-analyse een kwalijke zaak is. In plaats daarvan zouden we willen beredeneren dat we de sleutel van Meilof kunnen gebruiken als die dezelfde waarde heeft als sleutel van Geeft.[5]

[5] En bovendien, dat Meilof en Geert mogelijk dezelfde persoon zijn als ze kennelijk dezelfde sleutel gebruiken...

De belangrijkste technische bijdrage in dit proefschrift is dan ook dat we laten zien, hoe je tegelijkertijd over de betekenis én de waarde van informatie kunt redeneren. Laten we ons voorbeeldje van symmetrische encryptie er weer bij halen. We gaan weer redeneerregels opstellen voor de uitspraak "als een aanvaller lijst $\mathcal{C}$ van berichten heeft gezien, dan kent hij bericht X", maar nu rekening houdend met betekenis én waarde. Berichten zien er hetzelfde uit als eerst, dus bijvoorbeeld $\mathsf{enc}(\mathsf{geheim}, \mathsf{sleutel\_meilof})$ en $\mathsf{sleutel\_geert}$. Maar nu modelleren we ook welke berichten dezelfde waarde hebben. We gebruiken symbool $\doteq$, dus bijvoorbeeld $\mathsf{sleutel\_meilof} \doteq \mathsf{sleutel\_geert}$. Onze eerste redeneerregel blijft hetzelfde, maar we passen onze tweede redeneerregel aan om rekening

$$\dfrac{\dfrac{}{\mathcal{C} \vdash \mathsf{enc}(\mathsf{geheim},\mathsf{sleutel\_meilof})}\,(\vdash 1) \quad \dfrac{}{\mathcal{C} \vdash \mathsf{sleutel\_geert}}\,(\vdash 1)}{\mathcal{C} \vdash \mathsf{geheim}}\,(\vdash 2')$$

Figuur 2: Formele afleiding van $\mathcal{C} \vdash$ geheim met onderscheid van betekenis en waarde: we mogen $(\vdash 2')$ toepassen omdat sleutel_meilof $\doteq$ sleutel_geert

te houden met de waarde van berichten:

Als X in de lijst $\mathcal{C}$ voorkomt, dan kent de aanvaller X. $(\vdash 1)$

Als de aanvaller $\mathsf{enc}(X, Y)$ en Y' kent met $Y' \doteq Y$, dan kent hij ook X. $(\vdash 2')$

Stel nu dat een aanvaller de lijst met berichten

$$\mathcal{C} = \{\mathsf{enc}(\mathsf{geheim}, \mathsf{sleutel\_meilof}), \mathsf{sleutel\_geert}\}$$

heeft gezien. We kunnen met de twee bovenstaande redeneerregels afleiden dat hij nu geheim kent. De redenering, die we laten zien in Figuur 2, is bijna hetzelfde als die in Figuur 1, behalve dat we nu regel $(\vdash 2')$ toepassen met sleutel_geert, waarbij we op de achtergrond gebruik maken van het feit dat sleutel_meilof $\doteq$ sleutel_geert.

Deze twee regels zijn nog niet genoeg voor een volledig redeneersysteem. We hebben bijvoorbeeld in de bovenstaande redenering sleutel_geert gebruikt alsof het sleutel_meilof is. Maar als dit lukt en het ontcijferen van de encryptie levert wat op, dan weten we ook dat sleutel_meilof en sleutel_geert hetzelfde waren. Om met dit soort aspecten rekening te houden, moeten we extra regels toevoegen. De voornaamste bijdrage van dit proefschrift is een complete verzameling regels die een goed beeld geeft van kennis van van betekenis en waarde van informatie.

Slot

Samenvattend introduceert dit proefschrift een nieuw redeneersysteem voor kennis van cryptografische berichten. Met dit systeem kun je privacy (in de zin van kennis over persoonlijke informatie) in computersystemen analyseren. Om het pratisch nut van het systeem aan te tonen, hebben we in dit proefschrift twee daadwerkelijke grote privacy-analyses uitgevoerd. Bovendien hebben we, om de theoretische kracht van het systeem aan te tonen, wiskundig laten zien dat het bestaande ad-hoctechnieken voor privacy-analyse veralgemeniseert. Al met al draagt dit proefschrift hiermee hopelijk bij aan een beter begrip van privacy-aspecten van informatieuitwisseling in de gevaarlijke wereld van het internet.

Appendix B
Important Dates

January 4, 2010 First working day in Security group

June 23–24, 2010 Attended *Interdisciplinary Privacy Course*, K.U. Leuven, Leuven, **Belgium**

August 2–6, 2010 Attended *PrimeLife/IFIP Summer School 2010*, Helsingborg, **Sweden**

August 28, 2010 Participated in half-marathon, Groningen, **The Netherlands** (finishing in 1:53:36)

September 9, 2010 Presented at the EiPSI seminar, Eindhoven, **The Netherlands** (presentation title: "Modeling identity-related properties and their privacy strength")

September 16–17, 2010 Attended and presented at the *7th International Workshop on Formal Aspects of Security & Trust*, Pisa, **Italy** (presentation title: "Modeling identity-related properties and their privacy strength"; Figure 1)

Sept 26–Oct 01, 2010 Attended *Summer School on Applied Cryptographic Protocols*, Mykonos, **Greece** (Figure 2)

October 10, 2010 Participated in half-marathon, Eindhoven, **The Netherlands** (finishing in 1:57:19)

November 18–19, 2010 Attended, gave flash presentation, and presented poster at STW.ICT, Veldhoven, **The Netherlands** (poster title: "How do I manage my on-line identity with my mobile phone?" & "Modeling identity-related properties and their privacy strength")

May 13, 2011 Presented at *Crypto Working Group* meeting, Utrecht, **The Netherlands** (presentation title: "Formal privacy analysis for communication protocols")

June 27–July 4, 2011 Organised and participated in *The First EiPSI Open Tennis Championships*, TU/e, **The Netherlands** (going out in the quarter-finals against Jerry den Hartog 6-3 6-1)

August 28, 2011 Participated in half-marathon, Groningen, **The Netherlands** (finishing in 1:52:13)

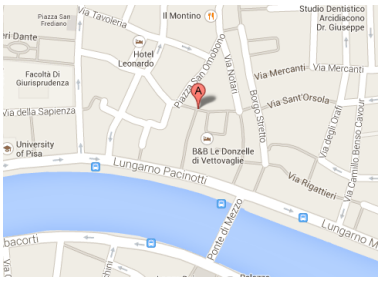


Figure 1: A good restaurant in Pisa

Figure 2: The swimming pool and view from the Saint John resort on Mykonos (PICTURE: PETER VAN LIESDONK)

Figure 3: Finish of the 2011 Eindhoven half-marathon

October 9, 2011 Participated in half-marathon, Eindhoven, **The Netherlands** (finishing in 1:52:31; Figure 3)

October 25, 2011 Visited the Bridge World Championships, Veldhoven, **The Netherlands** (Figure 4)

November 14–15, 2011 Attended, gave flash presentation, and presented poster at the ICT.OPEN 2011, Veldhoven, **The Netherlands** (poster title: "Formal Privacy Analysis of Communication Protocols for Identity Management")

November 16, 2011 Presented at the EiPSI seminar, Eindhoven, **The Netherlands** (presentation title: "Formal Privacy Analysis of Communication Protocols for Identity Management")

December 15–19, 2011 Attended and presented at the *7th International Conference on Information Systems Security*, Kolkata, **India** (presentation title: "Formal Privacy Analysis of Communication Protocols for Identity Management"; Figure 5)

February 16, 2012 Eilandje 1 outing to Concert Carnavalesk (Figure 6)

August 20–27, 2012 Organised and participated in *The Second EiPSI Open Tennis Championships*, TU/e, **The Netherlands** (losing to Iason Zisis in the first round, 6-3 6-4)

September 10–12, 2012 Attended the *17th European Symposium on Research in Computer Security*, Pisa, **Italy** (Figure 1)

September 13–14, 2012 Attended and presented at the *8th International Workshop on Security and Trust Management*, Pisa, **Italy** (presentation title: "Formal Modelling of (De)Pseudonymisation: A Case Study in Health Care Privacy ")

September 19, 2012 Presented at the EiPSI seminar, Eindhoven, **The Netherlands** (presentation title: "Formal Modelling of (De)Pseudonymisation: A Case Study in Health Care Privacy")

October 14, 2012 Participated in half-marathon, Eindhoven, **The Netherlands** (finishing in 1:48:33)

October 22–23, 2012 Attended, presented, and presented poster at the ICT.OPEN 2012, Rotterdam, **The Netherlands** (presentation title: "Formal Modelling of (De)Pseudonymisation", poster title: "Is Privacy By Data Minimisation Satisfied by Existing Systems?")

June 3–7, 2013 Attended and presented at the *7th IFIP WG 11.11 International Conference on Trust Management*, Malaga, **Spain** (presentation title: "Symbolic Privacy Analysis through Linkability and Detectability")

Aug 26–30, 2013 Organised and participated in *The Third EiPSI Open Tennis Championships*, TU/e, **The Netherlands** (beaten in first round of main tournament by Berry Schoenmakers, 6-4 6-3;

Figure 4: Table on which one of the semi-finals of the 2011 Bridge World Championships was played

Figure 5: "Relax" watch purchased in Kolkata

Figure 6: Outfits for Concert Carnavalesk

winner of fair-play award, beating Antonino Simone in the final, 6-1 7-5: Figure 7)

September 9–11, 2013 Attended the *18th European Symposium on Research in Computer Security*, Egham, **United Kingdom**

September 12–13, 2013 Attended and presented at the *8th DPM International Workshop on Data Privacy Management*, Egham, **United Kingdom** (presentation title: "Are On-Line Personae Really Unlinkable?")

October 13, 2013 Participated in half-marathon, Eindhoven, **The Netherlands** (finishing in 1:48:54)

November 4–8, 2013 Attended and presented poster at the *20th ACM Conference on Computer and Communications Security*, Berlin, **Germany** (poster title: "TRIPLEX: Verifying Data Minimisation in Communication Systems")

November 18, 2013 Presented at the EiPSI seminar, Eindhoven, **The Netherlands** (presentation title: "Privacy by Representation?")

November 27–28, 2013 Attended and presented poster at the ICT.OPEN 2013, Eindhoven, **The Netherlands** (poster title: "TRIPLEX: Verifying Data Minimisation in Communication Systems")

December 24, 2013 Last working day in Security group

Figure 7: The author, holding the fair-play trophy of the 2013 EiPSI Open (PICTURE: DION BOESTEN)

Distance	Date	Time
100m	17-01-12	13,69
500m	03-12-13	54,50
1000m	17-11-13	1:51,50
1500m	03-12-13	2:57,38
3000m	08-01-13	6:20,19
(Hour)	05-02-13	66 laps

Table 10: Personal best speed skating times 2010–2013 (SOURCE: ESSV ISIS)

Appendix C
Summary

Objective Privacy
Understanding the Privacy Impact of Information Exchange

Distributed systems are software systems in which components, often run by different organisations, communicate and coordinate their actions by passing messages over a network. This message passing over a network (typically, the internet) is done using communication protocols prescribing what information should be exchanged, in what order, and using what format. Over time, the design of such communication protocols has proven to be a subtle and error-prone task.

In particular, when these communication protocols are used to exchange personal information between different organisations, privacy is a major concern. Protocols involving personal information are getting more and more prevalent, e.g. in e-voting, smart metering, and identity management. Organisations are legally obliged to minimise the amount of personal information they deal with; so in particular, they need to use communication protocols designed with privacy in mind.

However, in the literature, there is no satisfactory way to obtain a good understanding of the privacy impact of using particular communication protocols. Existing comparisons are typically performed in a high-level and imprecise way. Precisely analysing the underlying cryptographic primitives is too technical, and hard to automate. It is possible to perform precise analyses by abstracting away from cryptography using formal methods, but this requires a formal encoding of privacy aspects. Today, such encodings are ad-hoc, and difficult to perform automated analysis on.

In this thesis, we propose techniques for obtaining a rigorous understanding of the privacy impact of information exchange in distributed systems. These techniques are designed to provide precise and verifiable results that are easy to interpret, but also to be largely automated. As a first step, we have proposed a model in which the knowledge of actors about personal information can be precisely expressed. This model is high-level, but can nonetheless be used to precisely capture a wide range of privacy concerns.

As a second step, we show how this model is combined with existing formal methods techniques for modelling cryptographic primitives. Namely, we have presented three different ways of automatically deciding privacy properties based on a formal model of information exchange. The first method, based on inference rules, can be used to efficiently analyse systems that use the most common cryptographic primitives. The second method, based on equational theories, allows more cryptographic primitives to be modelled at a higher level of detail, but is less efficient. The third method, a variant of the first, can be used to determine what conditions should be satisfied for privacy to be guaranteed, instead of whether privacy is guaranteed in a particular situation. This gives more general results, but the method is less efficient and the results are harder to interpret.

Finally, we have demonstrated the feasibility of applying our methods to actual systems by performing two concrete case studies. We have compiled a comprehensive set of privacy requirements for identity management, and we have analysed four different identity management systems against these requirements. Also, we have performed a rigorous analysis of achievable privacy guarantees for patient data pseudonymisation. In conclusion, the tools developed in this thesis have given us new ways to understand how the use of different communication protocols influences what privacy-sensitive information the actors in a system learn.

Appendix D
Curriculum Vitae

Meilof Veeningen was born on August 23, 1985 in Utrecht, The Netherlands. From 1997 to 2003, he did his pre-university education at the Stedelijk Gymnasium Leeuwarden. In 2003, he was selected as one of 6 Dutch high school students to represent The Netherlands at the International Mathematical Olympiad in Tokyo, Japan.

After finishing his pre-university education, he studied Mathematics and Computer Science at the University of Groningen. In 2004, he got the "Young Talent Encouragement Award", awarded by the Stieltjes Institute for Mathematics for getting the highest average grade on his first year diploma of all mathematics students at his university. He obtained his BSc degrees in Mathematics and Computer Science in 2007; the degree in Mathematics was awarded cum laude. In 2007, he visited the University of Bristol as an Erasmus exchange student. In 2009, he obtained his MSc degree in Mathematics, graduating cum laude with Jaap Top on his thesis "Invariants under simultaneous conjugation of SL(2) matrices". In 2006-2007, he was treasurer of the board of student society "Cleopatra A.S.G". He was also a founding and honorary member of the "Inchperfect" snooker society.

In 2010, he started a PhD project at Eindhoven University of Technology. His work was part of Mobiman, a research project on identity management on mobile devices supported by the Dutch Sentinels research program. The results of this work are presented in this dissertation.

Since 2014, he is working as a researcher at Eindhoven University of Technology, working on Secure Multi-Party Computation under Berry Schoenmakers in the Coding Theory and Cryptology Group.

Appendix E
Acknowledgements

I gratefully acknowledge the help I got during this project.

First, many thanks go out to my supervisory team. Many thanks, first, to Nicola, for always being there to help (or to watch Champions League), and for offering a hassle-free path towards the PhD defense. Many thanks, also, to Benne, for providing the mathematicians' perspective. Also, thanks to Sandro for being my promotor, and for being involved in the latter stages of the project.

Thanks to Wim Hesselink, Berry Schoenmakers, Catuscia Palamidessi, and Pierangela Samarati for taking place in my PhD committee; for assessing my thesis (I wanted to write a shorter thesis than this, sorry); and in particular, to the external members for taking the trouble of travelling to Eindhoven for my defense. Thanks to Emile Aarts for chairing the committee, and to Marchien and Marianne for being my paranimfen.

Thanks to Gergely Alpár, Martijn Oostdijk, Ton van Opstal and Maarten Wegdam for being part of the Mobiman project, in the context of which this research was carried out.

Thanks to Evert Jan Evers, Hans Maring, and Léon Haszing, for providing useful feedback on the patient data pseudonymisation case study in Chapter 8.

Thanks to many colleagues at Eindhoven University of Technology for providing a nice working environment. Thanks to Dion and Sebastiaan, my fellow members of "Eilandje 1", for the great working visits. Thanks also to Peter and Relinde for the nice coffee breaks. Thanks to Henk for organising the daily tea breaks, which really helped me feel at home in EiPSI. Thanks to Thijs for many interesting discussions. Thanks to everybody else at EiPSI: Andreas, Anita, Antonino, Boris, Bruno, Chitchanok, Christiane, Dan, Daniel, Elisa, Fatih, Gaetan, Jan-Jaap, Jerry, Jing, Jolande, Jose, Mayla, Michael, Mikkel, Milan, Niels, Omer, Peter vL, Peter S, Ruben, Ruud, Sokratis, Tanja, Tony, and Wil. Thanks to everybody else at Mathematics and Computer Science with whom I shared coffee breaks, games of bridge, basketball or poker, drinks, colloquia, etcetera.

Finally, thanks to my parents, family and friends, and especially to Marleen, for their advise and support.

Bibliography

Publications Covered by this Thesis

Meilof Veeningen, Benne de Weger, and Nicola Zannone. Modeling identity-related properties and their privacy strength. In *Proceedings of the 7th International Workshop on Formal Aspects of Security & Trust (FAST '10)*, LNCS 6561, pages 126–140. Springer, 2011a. *(Cited on page(s): 32)*

Meilof Veeningen, Benne de Weger, and Nicola Zannone. Formal privacy analysis of communication protocols for identity management. In *Proceedings of the 7th International Conference on Information Systems Security (ICISS '11)*, LNCS 7093, pages 235–249. Springer, 2011b. *(Cited on page(s): 32, 56)*

Meilof Veeningen, Benne de Weger, and Nicola Zannone. Formal modelling of (de)pseudonymisation: A case study in health care privacy. In *Proceedings of the 8th Workshop on Security and Trust Management (STM '12)*, LNCS 7783, pages 145–160. Springer, 2012. *(Cited on page(s): 32, 169)*

Meilof Veeningen, Benne de Weger, and Nicola Zannone. Symbolic privacy analysis through linkability and detectability. In *Proceedings of the 7th International Conference on Trust Management (IFIPTM '13)*, AICT 401, pages 1–16. Springer, 2013b. *(Cited on page(s): 106, 111)*

Meilof Veeningen, Benne de Weger, and Nicola Zannone. Data minimisation in communication protocols: A formal analysis framework and application to identity management. *International Journal of Information Security*, 2014. DOI: 10.1007/s10207-014-0235-z. *(Cited on page(s): 32, 56, 114, 118, 122, 125, 154)*

Other Publications by the Author

Meilof Veeningen, Mayla Brusò, Jerry den Hartog, and Nicola Zannone. POSTER: TRIPLEX: Verifying data minimisation in communication systems. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS '13)*, pages 1415–1418. ACM, 2013a. *(Cited on page(s): 189)*

Meilof Veeningen, Antionio Piepoli, and Nicola Zannone. Are on-line personae really unlinkable? In *Proceedings of the 8th Data Privacy Management Workshop (DPM '13)*, LNCS 8247, pages 369–379. Springer, 2014. *(Cited on page(s): 189)*

Cited Publications

Martín Abadi. Secrecy by typing in security protocols. *Journal of the ACM*, (5):749–786, 1998. *(Cited on page(s): 11)*

Martín Abadi and Bruno Blanchet. Analyzing security protocols with secrecy types and logic programs. *Journal of the ACM*, (1):102–146, 2005. *(Cited on page(s): 61)*

Martín Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In *Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of programming languages (POPL '01)*, pages 104–115. ACM, 2001. *(Cited on page(s): 11, 52, 57, 173, 174)*

Gergely Alpár, Jaap-Henk Hoepman, and Johanneke Siljee. The identity crisis: Security, privacy and usability issues in identity management. arXiv:1101.0427 [cs.CR], 2011. *(Cited on page(s): 127, 129, 130, 133, 154)*

Ross Anderson. Can we fix the security economics of federated authentication? In *Proceedings of the 19th International Workshop on Security Protocols (SPW '11)*, LNCS 7114, pages 25–32. Springer, 2011. *(Cited on page(s): 154)*

Myrto Arapinis, Tom Chothia, Eike Ritter, and Mark Ryan. Analysing unlinkability and anonymity using the applied pi calculus. In *Proceedings of the 2010 23rd IEEE Computer Security Foundations Symposium (CSF '10)*. IEEE, 2010. *(Cited on page(s): 11, 12, 52, 176, 177, 179, 181, 186)*

Myrto Arapinis, Sergiu Bursuc, and Mark Ryan. Privacy supporting cloud computing: Confichair, a case study. In *Proceedings of the First Conference on Principles of Security and Trust (POST '12)*, LNCS 7215, pages 89– 108. Springer, 2012. *(Cited on page(s): 12, 52, 176, 178, 180, 181, 184, 186)*

Claudio Agostino Ardagna, Jan Camenisch, Markulf Kohlweiss, Ronald Leenes, Gregory Neven, Bart Priem, Pierangela Samarati, Dieter Sommer, and Mario Verdicchio. Exploiting cryptography for privacy-enhanced access control: A result of the prime project. *Journal of Computer Security*, 18 (1):123–160, 2010a. *(Cited on page(s): 154)*

Claudio Agostino Ardagna, Sabrina De Capitani di Vimercati, Sara Foresti, Stefano Paraboschi, and Pierangela Samarati. Minimizing disclosure of private information in credential-based interactions: A graph-based approach. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing, SocialCom / IEEE International Conference on Privacy, Security, Risk and Trust (PASSAT '10)*, pages 743–750. IEEE, 2010b. *(Cited on page(s): 154)*

Alessandro Armando, David Basin, Yohan Boichut, Yannick Chevalier, Luca Compagna, Jorge Cuellar, Paul Hankes Drielsma, Pierre-Cyrille Heám, Jacopo Mantovani, Sebastian Mödersheim, David von Oheimb, Michaël Rusinowitch, Judson Santiago, Mathieu Turuani, Luca Viganò, and Laurent Vigneron. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In *Proceedings of the 17th International Conference on Computer Aided Verification (CAV'05)*, LNCS 3576, pages 281–285. Springer, 2005. *(Cited on page(s): 11, 174, 176)*

Alessandro Armando, Roberto Carbone, Luca Compagna, Jorge Cuellar, and Llanos Tobarra Abad. Formal analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps. In *Proceedings of the 6th ACM Workshop on Formal Methods in Security Engineering (FMSE '08)*, pages 1–10. ACM, 2008. *(Cited on page(s): 155, 176)*

Michael Backes, Matteo Maffei, and Dominique Unruh. Zero-knowledge in the applied pi-calculus and automated verification of the direct anonymous attestation protocol. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy (SSP '08)*, pages 202–215. ACM, 2008. *(Cited on page(s): 58, 83, 122)*

Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008. *(Cited on page(s): 174)*

Endre Bangerter, Jan Camenisch, and Anna Lysyanskaya. A cryptographic framework for the controlled release of certified data. In *Proceedings of the 12th International Workshop on Security Protocols (SPW '04)*, LNCS 3957, pages 20–42. Springer, 2004. *(Cited on page(s): 10, 122, 123, 124, 127, 129, 130, 131, 132, 133, 135, 136, 141, 142)*

Mathieu Baudet, Bogdan Warinschi, and Martín Abadi. Guessing attacks and the computational soundness of static equivalence. *Journal of Computer Security*, (5):909–968, 2010. *(Cited on page(s): 53, 83)*

Donald Beaver. Secure multiparty protocols and zero-knowledge proof systems tolerating a faulty minority. *Journal of Cryptology*, 4(2):75–122, 1991. *(Cited on page(s): 10, 182)*

Mihir Bellare. Practice-oriented provable security. In *Proceedings of First International Workshop on Information Security (ISW '97)*, LNCS 1396, pages 1–15. Springer, 1998. *(Cited on page(s): 10)*

Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, (4):469–491, 2008. *(Cited on page(s): 55, 57)*

Abhilasha Bhargav-Spantzel, Jan Camenisch, Thomas Gross, and Dieter Sommer. User centricity: A taxonomy and open issues. *Journal of Computer Security*, (5):493–527, 2007a. *(Cited on page(s): 129)*

Abhilasha Bhargav-Spantzel, Anna Cinzia Squicciarini, Matthew Young, and Elisa Bertino. Privacy requirements in identity management solutions. In *Proceedings of the IEEE International Workshop on Human Computer Interaction 2007 (HCI '07)*, LNCS 4558, pages 694–702. Springer, 2007b. *(Cited on page(s): 127, 130, 131, 132, 133, 155)*

Indrajit Bhattacharya and Lise Getoor. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data*, (1), 2007. *(Cited on page(s): 187, 189)*

Mikhail Bilenko, Sugato Basu, and Mehran Sahami. Adaptive product normalization: Using online learning for record linkage in comparison shopping. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM '05)*, pages 58–65. IEEE, 2005. *(Cited on page(s): 187, 189)*

Bruno Blanchet. Automatic proof of strong secrecy for security protocols. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy (S&P '04)*, pages 86–100. IEEE, 2004. *(Cited on page(s): 11, 174, 175, 176, 185, 188)*

Bruno Blanchet and Ben Smyth. Proverif 1.85: Automatic cryptographic protocol verifier, user manual and tutorial. `http://prosecco.gforge.inria.fr/personal/bblanche/proverif/`, 2011. *(Cited on page(s): 11, 12, 43, 53, 83, 174, 176)*

Bruno Blanchet, Martín Abadi, and Cédric Fournet. Automated verification of selected equivalences for security protocols. *Journal of Logic and Algebraic Programming*, (1):3–51, 2008. *(Cited on page(s): 11, 12, 52, 58, 83, 84, 188)*

Jens-Matthias Bohli and Andreas Pashalidis. Relations among privacy notions. *ACM Transactions on Information and System Security*, (1):4:1–4:24, 2011. *(Cited on page(s): 178, 187)*

Michele Boreale. Symbolic trace analysis of cryptographic protocols. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP '01)*, LNCS 2076, pages 667–681. Springer, 2001. *(Cited on page(s): 173, 174, 185)*

Laura Brandimarte, Alessandro Acquisti, and George Loewenstein. Misplaced confidences: Privacy and the control paradox. Ninth Workshop on the Economics of Information Security (WEIS '10), 2010. *(Cited on page(s): 155)*

James Brown, Phil Stradling, and Craig H. Wittenberg. U-Prove CTP R2 white paper. `http;//research.microsoft.com/en-us/projects/u-prove/`, 2010. *(Cited on page(s): 129)*

Mayla Brusó, Konstantinos Chatzikokolakis, Sandro Etalle, and Jerry den Hartog. Linking unlinkability. In *Proceedings of the Seventh International Symposium on Trustworthy Global Computing (TGC '12)*, LNCS 8191, pages 129–144. Springer, 2013a. *(Cited on page(s): 178)*

Mayla Brusó, Konstantinos Chatzikokolakis, Sandro Etalle, and Jerry Hartog. Dissecting unlinkability. To appear, 2013b. *(Cited on page(s): 178)*

Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems*, pages 18–36, 1990. *(Cited on page(s): 143)*

Denis Butin and Giampaolo Bella. Verifying privacy by little interaction and no process equivalence. In *Proceedings of the International Conference on Security and Cryptography (SECRYPT '12)*, pages 251–256. SciTePress, 2012. *(Cited on page(s): 177)*

Denis Butin, David Gray, and Giampaolo Bella. Towards verifying voter privacy through unlinkability. In *Proceedings of the 5th International Symposium on Engineering Secure Software and Systems (ESSoS '13)*, LNCS 7781, pages 91–106. Springer, 2013. *(Cited on page(s): 177)*

Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Proceedings of EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT '01)*, LNCS 2045, pages 93–118. Springer, 2001. *(Cited on page(s): 182)*

Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *Proceedings of the 3rd International Conference on Security in Communication Networks (SCN '02)*, LNCS 2576, pages 268–289. Springer, 2003. *(Cited on page(s): 123, 125)*

Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Proceedings of the 24rd Annual International Cryptology Conference (CRYPTO '04)*, LNCS 3152, pages 56–72. Springer, 2004. *(Cited on page(s): 125)*

Jan Camenisch, Dieter Sommer, and Roger Zimmermann. A general certification framework with applications to privacy-enhancing certificate infrastructures. In *Proceedings of the IFIP TC-11 21st International Information Security Conference (SEC '06)*, IFIP 201, pages 25–37. Springer, 2006. *(Cited on page(s): 151)*

Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography (PKC '09)*, LNCS 5443, pages 481–500. Springer, 2009. *(Cited on page(s): 151)*

Jan Camenisch, Sebastian Mödersheim, and Dieter Sommer. A formal model of identity mixer. In *Proceedings of the 15th International Workshop on Formal Methods for Industrial Critical Systems (FMICS'10)*, LNCS 6371, pages 198–214. Springer, 2010. *(Cited on page(s): 122, 125, 142, 155)*

Kim Cameron. The laws of identity. `http://www.identityblog.com/?p=352`, 2006. *(Cited on page(s): 154)*

Jean Camp. Identity management's misaligned incentives. *IEEE Security & Privacy*, (6):90–94, 2010. *(Cited on page(s): 154)*

Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS '01)*, pages 136–145. IEEE, 2001. *(Cited on page(s): 10)*

Scott Cantor, John Kemp, Rob Philpott, and Eve Maler (eds.). Assertions and protocols for the OASIS security assertion markup language (SAML) v2.0. `http://saml.xml.org/saml-specifications`, 2005. *(Cited on page(s): 145, 155)*

David Chadwick and George Inman. Attribute aggregation in federated identity management. *IEEE Computer*, (5):33–40, 2009. *(Cited on page(s): 10, 127, 129, 130, 131, 132, 133, 134, 135, 149)*

Konstantinos Chatzikokolakis. *Probabilistic and Information-Theoretic Approaches to Anonymity*. PhD thesis, Laboratoire d'Informatique (LIX), École Polytechnique, Paris, 2007. *(Cited on page(s): 187)*

Surajit Chaudhuri, Venkatesh Ganti, and Rajeev Motwani. Robust identification of fuzzy duplicates. In *Proceedings of the 21st International Conference on Data Engineering (ICDE '05)*, pages 865–876. IEEE, 2005. *(Cited on page(s): 187, 189)*

David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, (2):84–90, 1981. *(Cited on page(s): 187)*

David Chaum and Eugène van Heyst. Group signatures. In *Proceedings of EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques (EUROCRYPT '91)*, LNCS 547, pages 257–265. Springer, 1991. *(Cited on page(s): 149)*

Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS '95)*, pages 41–50. IEEE, 1995. *(Cited on page(s): 150)*

Ştefan Ciobâcă, Stéphanie Delaune, and Steve Kremer. Computing knowledge in security protocols under convergent equational theories. In *Proceedings of the 22nd International Conference on Automated Deduction (CADE'09)*, LNCS 5663, pages 355–370. Springer, 2009. *(Cited on page(s): 58, 75, 83, 84)*

Brecht Claerhout and Georges De Moor. Privacy protection for clinical and genomic data: The use of privacy-enhancing techniques in medicine. *International Journal of Medical Informatics*, (2-4):257–265, 2005. *(Cited on page(s): 169)*

Edmund M. Clarke, Somesh Jha, and Wilfredo R. Marrero. Using state space exploration and a natural deduction style message derivation engine to verify security protocols. In *Proceedings of the IFIP TC2/WG2.2,2.3 International Conference on Programming Concepts and Methods (PROCOMET '98)*, pages 87–106. Chapman & Hall, 1998. *(Cited on page(s): 174, 185)*

Hubert Comon-Lundh, Véronique Cortier, and Eugen Zălinescu. Deciding security properties for cryptographic protocols. application to key cycles. *ACM Transactions on Computational Logic*, (2), 2010. *(Cited on page(s): 185)*

Luca Compagna, Paul El Khoury, Alzbeta Krausová, Fabio Massacci, and Nicola Zannone. How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns. *Artificial Intelligence and Law*, (1):1–30, 2009. *(Cited on page(s): 154, 187)*

Ricardo Corin, Sreekanth Malladi, Jim Alves-Foss, and Sandro Etalle. Guess what? here is a new tool that finds some new guessing attacks (extended abstract). Third Workshop on Issues in the Theory of Security, 2003. *(Cited on page(s): 176)*

Ricardo Corin, Jeroen Doumen, and Sandro Etalle. Analysing password protocol security against off-line dictionary attacks. *Electronic Notes in*

Theoretical Computer Science, pages 47–63, 2005. *(Cited on page(s): 54, 57, 58, 64, 65, 83, 175, 185)*

Véronique Cortier, Stéphanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, (1):1–43, 2006. *(Cited on page(s): 54)*

Véronique Cortier, Heinrich Hördegen, and Bogdan Warinschi. Explicit randomness is not necessary when modeling probabilistic encryption. *Electronic Notes in Theoretical Computer Science*, pages 49–65, 2007. *(Cited on page(s): 53, 83)*

George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair. *Distributed Systems: Concepts and Design*. Addison-Wesley, 2005. *(Cited on page(s): 7)*

Ronald Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, Universiteit van Amsterdam, 1997. *(Cited on page(s): 118)*

Morten Dahl, Stéphanie Delaune, and Graham Steel. Formal analysis of privacy for anonymous location based services. In *Proceedings of the Joint Workshop on Theory of Security and Applications (TOSCA '11)*, LNCS 6993, pages 98–112. Springer, 2011. *(Cited on page(s): 83, 177, 180, 181, 184)*

Don Davis. Defective sign & encrypt in s/mime, pkcs#7, moss, pem, pgp, and xml. In *Proceedings of the General Track: 2001 USENIX Annual Technical Conference (USENIX '01)*, pages 65–78. USENIX, 2001. *(Cited on page(s): 8)*

Stéphanie Delaune, Steve Kremer, and Mark Ryan. Composition of password-based protocols. In *Proceedings of the 21st Computer Security Foundations Symposium (CSF '08)*, pages 239–251. IEEE, 2008. *(Cited on page(s): 58, 59, 61, 65, 175)*

Stéphanie Delaune, Steve Kremer, and Mark Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, (4): 435–487, 2009. *(Cited on page(s): 11, 52, 58, 62, 83, 84, 175, 177, 180, 181)*

Mina Deng, Danny De Cock, and Bart Preneel. Towards a cross-context identity management framework in e-health. *Online Information Review*, (3):422–442, 2009. *(Cited on page(s): 169)*

Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 243–320. MIT Press, 1990. *(Cited on page(s): 76)*

Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Fragments and loose associations: Respecting privacy in data publishing. *Proceedings of the VLDB Endowment*, 3 (1):1370–1381, 2010. *(Cited on page(s): 187)*

Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Authorization enforcement in distributed query evaluation. *Journal of Computer Security*, 19(4):751–794, 2011. *(Cited on page(s): 155)*

Sabrina De Capitani di Vimercati, Sara Foresti, Giovanni Livraga, and Pierangela Samarati. Data privacy: Definitions and techniques. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 20(6):793–818, 2012. *(Cited on page(s): 187)*

Sabrina De Capitani di Vimercati, Sarah Foresti, Sushil Jajodia, Giovanni Livraga, Stefano Paraboschi, and Pierangela Samarati. Fragmentation in presence of data dependencies. *IEEE Transactions on Dependable and Secure Computing*, To appear. *(Cited on page(s): 187)*

Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium (SEC '04)*, pages 303–320. USENIX, 2004. *(Cited on page(s): 187)*

Danny Dolev and Andrew Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, (2):198–208, 1981. *(Cited on page(s): 11, 33, 174)*

Naipeng Dong, Hugo Jonker, and Jun Pang. Formal analysis of privacy in an ehealth protocol. In *Proceedings of the 17th European Symposium on Research in Computer Security (ESORICS '12)*, LNCS 7459, pages 325–342. Springer, 2012. *(Cited on page(s): 12, 52, 83, 122, 176, 177, 179, 180, 181, 186)*

Naipeng Dong, Hugo Jonker, and Jun Pang. Enforcing privacy in the presence of others: Notions, formalisations and relations. In *Proceedings of the 18th European Symposium on Research in Computer Security (ESORICS '13)*, LNCS 8134, pages 499–516. Springer, 2013. *(Cited on page(s): 11, 12, 52, 175, 176, 180, 181, 184, 186)*

Jannik Dreier, Pascal Lafourcade, and Yassine Lakhnech. Defining privacy for weighted votes, single and multi-voter coercion. In *Proceedings of the 17th European Symposium on Research in Computer Security (ESORICS '12)*, LNCS 7459, pages 451–468. Springer, 2012. *(Cited on page(s): 175, 177)*

Jannik Dreier, Pascal Lafourcade, and Yassine Lakhnech. Formal verification of e-auction protocols. In *Proceedings of the 2nd Conference on Principles of Security and Trust (POST '13)*, LNCS 7796, pages 247–266. Springer, 2013. *(Cited on page(s): 52, 176, 177, 180, 181)*

Cynthia Dwork. Differential privacy. In *Proceedings of 33rd International Colloquium on Automata, Languages and Programming (ICALP '06)*, LNCS 4052, pages 1–12. Springer, 2006. *(Cited on page(s): 187)*

Marlena Erdos and Scott Cantor (eds.). The Shibboleth architecture. `http://shibboleth.internet2.edu/`, 2005. *(Cited on page(s): 127, 130)*

Ivan P. Fellegi and Alan B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, (328):1183–1210, 1969. *(Cited on page(s): 186)*

John Ferro, Lisa Singh, and Micah Sherr. Identifying individual vulnerability based on public data. In *Proceedings of the Eleventh Annual International Conference on Privacy, Security and Trust (PST '13)*, pages 119–126. IEEE, 2013. *(Cited on page(s): 189)*

George Fyffe. Insider threats: Addressing the insider threat. *Network Security*, (3):11–14, 2008. *(Cited on page(s): 130)*

Marit Hansen, Peter Berlich, Jan Camenisch, Sebastian Clauß, Andreas Pfitzmann, and Michael Waidner. Privacy-enhancing identity management. *Information Security Technical Report*, (1):35–44, 2004. *(Cited on page(s): 8, 127, 130, 131, 132, 133, 154, 155)*

Alejandro Hevia and Daniele Micciancio. An indistinguishability-based characterization of anonymous channels. In *Proceedings of the 8th International Symposium on Privacy Enhancing Technologies (PETS '08)*, LNCS 5134, pages 24–43. Springer, 2008. *(Cited on page(s): 178, 187)*

Jeff Hodges, John Kemp, Robert Aarts, Greg Whitehead, and Paul Madsen (eds.). Liberty ID-WSF SOAP binding specification. `http://projectliberty.org/`, 2006. *(Cited on page(s): 130, 145)*

Jaap-Henk Hoepman, Rieks Joosten, and Johanneke Siljee. Comparing identity management frameworks in a business context. In *Proceedings of the 4th IFIP WG 9.2, 9.6, 11.6, 11.7/FIDIS International Summer School (FIDIS '08)*, IFIP AICT 298, pages 184–196. Springer, 2008. *(Cited on page(s): 127, 154, 155)*

Dominic Hughes and Vitaly Shmatikov. Information hiding, anonymity and privacy: a modular approach. *Journal of Computer Security*, (1):3–36, 2004. *(Cited on page(s): 187)*

Hans Hüttel and Michael D. Pedersen. A logical characterisation of static equivalence. *Electronic Notes in Theoretical Computer Science*, pages 139–157, 2007. *(Cited on page(s): 59, 64)*

Independent Centre for Privacy Protection Schleswig-Holstein. Identity management systems (IMS): Identification and comparison study. `https://www.datenschutzzentrum.de`, 2003. *(Cited on page(s): 9, 127, 130, 154, 155)*

Audun Jøsang and Simon Pope. User-centric identity management, 2005. *(Cited on page(s): 129, 130)*

Sampo Kellomäki (ed.). TAS^3 architecture. `http://vds1628.sivit.org/tas3/`, 2009. *(Cited on page(s): 127)*

Martijn Koot. *Measuring and predicting anonymity*. PhD thesis, University of Amsterdam, 2012. *(Cited on page(s): 187, 189)*

Hanna Köpcke and Erhard Rahm. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering*, (2):197–210, 2010. *(Cited on page(s): 186)*

Susan Landau and Tyler Moore. Economic tussles in federated identity management. Tenth Workshop on the Economics of Information Security (WEIS '11), 2011. *(Cited on page(s): 154)*

Susan Landau, Hubert Gong, and Robin Wilton. Achieving privacy in a federated identity management system. In *Proceedings of the 13th International Conference on Financial Cryptography and Data Security (FC '09)*, LNCS 5628, pages 51–70. Springer, 2009. *(Cited on page(s): 127, 155)*

Lucie Langer, Hugo Jonker, and Wolter Pieters. Anonymity and verifiability in voting: Understanding (un)linkability. In *Proceedings of the 12th International Conference on Information and Communications Security (ICICS '10)*, LNCS 6476, pages 296–310. Springer, 2010. *(Cited on page(s): 178)*

Laurie Law, Alfred Menezes, Minghua Qu, Jerry Solinas, and Scott Vanstone. An efficient protocol for authenticated key agreement. *Designs, Codes and Cryptography*, pages 119–134, 2003. *(Cited on page(s): 142)*

Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In *Proceedings of International Conference on Data Engineering (ICDE '07)*, pages 106–115. IEEE, 2007. *(Cited on page(s): 187)*

Xiangxi Li, Yu Zhang, and Yuxin Deng. Verifying anonymous credential systems in applied pi calculus. In *Proceedings of the 8th International Conference on Cryptology and Network Security (CANS '09)*, LNCS 5888, pages 209–225. Springer, 2009. *(Cited on page(s): 125, 155)*

Gavin Lowe. Breaking and fixing the needham-schroeder public-key protocol using fdr. In *Proceedings of the Second International Workshop on Tools and Algorithms for Construction and Analysis of Systems (TACAS '96)*, LNCS 1055, pages 147–166. Springer, 1996. *(Cited on page(s): 143)*

Gavin Lowe. Analysing protocol subject to guessing attacks. *Journal of Computer Security*, (1):83–98, 2004. *(Cited on page(s): 176)*

Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, (1), 2007. *(Cited on page(s): 187)*

Catherine Meadows. Formal methods for cryptographic protocol analysis: Emerging issues and trends. *IEEE Journal on Selected Areas in Communications*, (1):44–54, 2003. *(Cited on page(s): 20)*

Alfred J. Menezes, Scott A. Vanstone, and Paul C. van Oorschot. *Handbook of Applied Cryptography*. CRC Press, 1996. *(Cited on page(s): 8, 42, 43)*

Robin Milner. *Communicating and Mobile Systems: the π-Calculus*. Cambridge University Press, 1999. *(Cited on page(s): 173)*

Sebastian A. Mödersheim, Thomas Groß, and Luca Vigan. Defining privacy is supposed to be easy. In *Proceedings of the 19th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-19)*, LNCS 8312, pages 619–635. Springer, 2013. *(Cited on page(s): 182)*

N. Mray, J.B. Reitsma, A.C.J. Ravelli, and G.J. Bonsel. Probabilistic record linkage is a valid and transparent tool to combine databases without a patient identification number. *Journal of Clinical Epidemiology*, (9):883–891, 2007. *(Cited on page(s): 187, 189)*

Arun Nanda. A technical reference for the information card profile v1.0. `http://msdn.microsoft.com/en-us/library/bb298802.aspx`, 2007. *(Cited on page(s): 129)*

Gregory Neven and Franz-Stefan Preiss. Attribute predicate profile of SAML and XACML. `http://markmail.org/message/2dha2sqmgni7wpc5`, 2011. *(Cited on page(s): 148)*

OECD. OECD guidelines on the protection of privacy and transborder flows of personal data. `http://www.oecd.org/`, 2002. *(Cited on page(s): 158)*

Office of the Data Protection Commissioner (Ireland). Data protection guidelines on research in the health sector. `http://www.dataprotection.ie`, 2007. *(Cited on page(s): 169)*

Parag and Pedro Domingos. Multi-relational record linkage. In *Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining (MRDM '04)*, pages 31–48. ACM, 2004. *(Cited on page(s): 187, 189)*

Parelsnoer Initiatief. Programma van eisen intstellingen. `http://www.parelsnoer.org/`, 2008. *(Cited on page(s): 8, 157, 159, 160)*

Parelsnoer Initiatief. Architecture central infrastructure. `http://www.parelsnoer.org/`, 2009. *(Cited on page(s): 157, 159, 160, 164)*

Joon S. Park and Ravi Sandhu. Smart certificates: Extending x.509 for secure attribute services on the web. Proceedings of the 22nd National Information Systems Security Conference (NISSC '99), 1999. *(Cited on page(s): 131, 133, 134, 148)*

Andreas Pashalidis and Bernd Meyer. Linking anonymous transactions: The consistent view attack. In *Proceedings of the 6th International Workshop on Privacy Enhancing Technologies (PET '06)*, LNCS 4258, pages 384–392. Springer, 2006. *(Cited on page(s): 27, 132)*

Lawrence C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, (1-2):85–128, 1998. *(Cited on page(s): 143, 173, 174, 177)*

Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. `http://dud.inf.tu-dresden.de/Anon_Terminology.shtml`, 2009. *(Cited on page(s): 132, 154, 155, 178, 187)*

Klaus Pommerening and Michael Reng. Secondary use of the EHR via pseudonymisation. *Studies in health technology and informatics*, pages 441–446, 2004. *(Cited on page(s): 157, 159, 169)*

Tatiana Pontes, Gabriel Magno, Marisa A. Vasconcelos, Aditi Gupta, Jussara M. Almeida, Ponnurangam Kumaraguru, and Virgilio Almeida. Beware of what you share: Inferring home location in social networks. In *12th International Conference on Data Mining Workshops (ICDMW '12)*, pages 571–578. IEEE, 2012. *(Cited on page(s): 187)*

Catherine Quantin, David-Olivier Jaquet-Chiffelle, Gouenou Coatrieux, Eric Benzenine, and François-André Allaert. Medical record search engines, using pseudonymised patient identity: An alternative to centralised medical records. *International Journal of Medical Informatics*, (2):e6–e11, 2011. *(Cited on page(s): 169)*

Jean-Jacques Quisquater, Louis Guillou, Marie Annick, and Tom Berson. How to explain zero-knowledge protocols to your children. In *Proceedings of the 9th Annual International Cryptology Conference (CRYPTO '89)*, LNCS 435, pages 628–631. Springer, 1989. *(Cited on page(s): 118)*

Alfredo Rial and George Danezis. Privacy-preserving smart metering. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society (WPES '11)*, pages 49–60. ACM, 2011. *(Cited on page(s): 32, 182)*

Alfredo Rial, Josep Balasch, and Bart Preneel. A privacy-preserving buyer-seller watermarking protocol based on priced oblivious transfer. *IEEE Transactions on Information Forensics and Security*, (1):202–212, 2011. *(Cited on page(s): 182)*

Robert Richardson. Computer crime & security survey. `http://www.gocsi.com/`, 2008. *(Cited on page(s): 9)*

Bernhard Riedl, Thomas Neubauer, Gernot Goluch, Oswald Boehm, Gert Reinauer, and Alexander Krumboeck. A secure architecture for the pseudonymization of medical data. In *Proceedings of the Second International Conference on Availability, Reliability and Security (ARES '07)*, pages 318 –324. IEEE, 2007. *(Cited on page(s): 169)*

Mauricio Sadinle and Stephen E. Fienberg. A generalized fellegi-sunter framework for multiple record linkage with application to homicide record systems. arXiv:1205.3217 [stat.AP], 2012. *(Cited on page(s): 187, 189)*

Pierangela Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001. *(Cited on page(s): 187)*

Pierangela Samarati and Latanya Sweeney. Generalizing data to provide anonymity when disclosing information (abstract). In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS '98)*, page 188. ACM, 1998. *(Cited on page(s): 187)*

Emili Sapena, Lluís Padró, and Jordi Turmo. A graph partitioning approach to entity disambiguation using uncertain information. In *Proceedings of the 6th International Conference on Advances in Natural Language Processing (GoTAL '08)*, LNCS 5221, pages 428–439. Springer, 2008. *(Cited on page(s): 187, 189)*

B. Schmidt, S. Meier, C. Cremers, and D. Basin. Automated analysis of diffie-hellman protocols and advanced security properties. In *Proceedings of the 25th Computer Security Foundations Symposium (CSF '12)*, pages 78–94. IEEE, 2012. *(Cited on page(s): 11, 174, 176)*

Steve Schneider and Abraham Sidiropoulos. CSP and anonymity. In *Proceedings of the 4th European Symposium on Research in Computer Security (ESORICS '96)*, LNCS 2482, pages 198–218. Springer, 1996. *(Cited on page(s): 187)*

Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Proceedings of the 9th Annual International Cryptology Conference (CRYPTO '89)*, LNCS 435, pages 239–252. Springer, 1989. *(Cited on page(s): 120)*

Berry Schoenmakers. Cryptography 2 (2WC13) / Cryptographic Protocols 1 (2WC17): Lecture Notes Cryptographic Protocols. `http://www.win.tue.nl/~berry/`, 2014. Version 1.0. *(Cited on page(s): 120)*

Kent Seamons, Marianne Winslett, Ting Yu, Lina Yu, and Ryan Jarvis. Protecting privacy during on-line trust negotiation. In *Proceedings of the 2nd International Conference on Privacy Enhancing Technologies (PET '02)*, LNCS 2482, pages 249–253. Springer, 2003. *(Cited on page(s): 132)*

Thomas J. Smedinghoff. Federated identity management: Balancing privacy rights, liability risks, and the duty to authenticate. `http://ssrn.com/abstract=1471599`, 2009. *(Cited on page(s): 130)*

Ben Smyth, Mark Ryan, and Liqun Chen. Formal analysis of anonymity in direct anonymous attestation schemes. In *Proceedings of the 8th International Workshop on Formal Aspects of Security & Trust (FAST '11)*, LNCS 7140, pages 245–262. Springer, 2012. *(Cited on page(s): 122, 125)*

Dieter Sommer, Marco Casassa Mont, and Siani Pearson. PRIME architecture V3. `https://www.prime-project.eu/`, 2008. *(Cited on page(s): 127)*

Sarah Spiekermann and Lorrie Faith Cranor. Engineering privacy. *IEEE Transactions on Software Engineering*, (1):67–82, 2009. *(Cited on page(s): 130, 131)*

Sandra Steinbrecher and Stefan Köpsell. Modelling unlinkability. In *Proceedings of the Workshop on Privacy Enhancing Technologies (PET '13)*, LNCS 2760, pages 32–47. Springer, 2003. *(Cited on page(s): 187)*

Suriadi Suriadi. *Strengthening and formally verifying privacy in identity management systems*. PhD thesis, Queensland University of Technology, 2010. *(Cited on page(s): 155)*

Paul F. Syverson and Stuart G. Stubblebine. Group principals and the formalization of anonymity. In *Proceedings of the World Congress on Formal Methods in the Development of Computing Systems (FM '99)*, LNCS 1708, pages 814–833. Springer, 1999. *(Cited on page(s): 187)*

Wouter Teepe. Integrity and dissemination control in administrative applications through information designators. *Computer Systems Science & Engineering*, (5), 2005. *(Cited on page(s): 169)*

Rose Tinabo, Fredrick Mtenzi, and Brendan O'Shea. Anonymisation vs. pseudonymisation: Which one is most useful for both privacy protection and usefulness of e-healthcare data. In *Proceedings of the International Conference for Internet Technology and Secured Transactions (ICITST '09)*, pages 1–6. IEEE, 2009. *(Cited on page(s): 169)*

Wiem Tounsi, Nora Cuppens-Boulahia, Frédéric Cuppens, and Joaquin Garcia-Alfaro. Formal verification of a key establishment protocol for epc gen2 rfid systems: Work in progress. In *Proceedings of the 4th Canada-France MITACS Conference on Foundations and Practice of Security (FPS '11)*, LNCS 6888, pages 242–251. Springer, 2012. *(Cited on page(s): 176)*

Carmela Troncoso. *Design and analysis methods for privacy technologies*. PhD thesis, Katholieke Universiteit Leuven, 2011. *(Cited on page(s): 9)*

Ton van Deursen, Sjouke Mauw, and Saa Radomirovi. Untraceability of rfid protocols. In *Proceedings of the Second International Workshop on Information Security Theory and Practices (WISTP '08)*, LNCS 5019, pages 1–15. Springer, 2008. *(Cited on page(s): 176)*

Meilof Veeningen, Benne de Weger, and Nicola Zannone. Modeling identity-related properties and their privacy strength. In *Proceedings of the 7th International Workshop on Formal Aspects of Security & Trust (FAST '10)*, LNCS 6561, pages 126–140. Springer, 2011a. *(Cited on page(s): 32)*

Meilof Veeningen, Benne de Weger, and Nicola Zannone. Formal privacy analysis of communication protocols for identity management. In *Proceedings of the 7th International Conference on Information Systems Security (ICISS '11)*, LNCS 7093, pages 235–249. Springer, 2011b. *(Cited on page(s): 32, 56)*

Meilof Veeningen, Benne de Weger, and Nicola Zannone. Formal modelling of (de)pseudonymisation: A case study in health care privacy. In *Proceedings of the 8th Workshop on Security and Trust Management (STM '12)*, LNCS 7783, pages 145–160. Springer, 2012. *(Cited on page(s): 32, 169)*

Meilof Veeningen, Mayla Brusò, Jerry den Hartog, and Nicola Zannone. POSTER: TRIPLEX: Verifying data minimisation in communication systems. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS '13)*, pages 1415–1418. ACM, 2013a. *(Cited on page(s): 189)*

Meilof Veeningen, Benne de Weger, and Nicola Zannone. Symbolic privacy analysis through linkability and detectability. In *Proceedings of the 7th International Conference on Trust Management (IFIPTM '13)*, AICT 401, pages 1–16. Springer, 2013b. *(Cited on page(s): 106, 111)*

Meilof Veeningen, Antionio Piepoli, and Nicola Zannone. Are on-line personae really unlinkable? In *Proceedings of the 8th Data Privacy Management Workshop (DPM '13)*, LNCS 8247, pages 369–379. Springer, 2014. *(Cited on page(s): 189)*

Meilof Veeningen, Benne de Weger, and Nicola Zannone. Data minimisation in communication protocols: A formal analysis framework and application to identity management. *International Journal of Information Security*, 2014. DOI: 10.1007/s10207-014-0235-z. *(Cited on page(s): 32, 56, 114, 118, 122, 125, 154)*

Jan Vossaert, Jorn Lapon, Bart De Decker, and Vincent Naessens. User-centric identity management using trusted modules. In *Proceedings of the 7th European Workshop on Public Key Infrastructures, Services and Applications (EuroPKI '10)*, LNCS 6711, pages 155–170. Springer, 2011. *(Cited on page(s): 10, 127, 129, 130, 131, 132, 133, 136, 137, 147)*

Ning Zhang, Alan Rector, Iain Buchan, Qi Shi, Dipak Kalra, Jeremy Rogers, Carole Goble, Steve Walker, David Ingram, and Peter Singleton. A linkable identity privacy algorithm for HealthGrid. *Studies in health technology and informatics*, pages 234–245, 2005. *(Cited on page(s): 169)*

List of Symbols

Personal Information Model and view (Chapter 2)

$(\mathsf{O}^{\mathsf{ctx}}, \mathcal{O}^{\mathsf{inf}}, \mathsf{O}^{\mathsf{cnt}}, \leftrightarrows, \sigma, \tau)$	Personal Information (PI) Model, consisting of context personal items $\mathsf{O}^{\mathsf{ctx}}$, personal items $\mathcal{O}^{\mathsf{inf}}$, contents personal items $\mathsf{O}^{\mathsf{cnt}}$, related relation $\leftrightarrows$, context-to-information map σ, and information-to-contents map τ (cf. Information Model, Chapter 3), page 23
$\mathsf{O}^{\mathsf{ctx}} := \mathsf{D}^{\mathsf{ctx}} \cup \mathsf{I}^{\mathsf{ctx}}$	Context personal items (context layer), partitioned into context data items $\mathsf{D}^{\mathsf{ctx}}$ and context identifiers $\mathsf{I}^{\mathsf{ctx}}$, page 22
$\mathcal{O}^{\mathsf{inf}} := \mathcal{D}^{\mathsf{inf}} \cup \mathcal{I}^{\mathsf{inf}}$	Personal items (information layer), partitioned into data items $\mathcal{D}^{\mathsf{inf}}$ and identifiers $\mathcal{I}^{\mathsf{inf}}$, page 21
$id\vert^{\pi}_{cli}$	Context item with domain π, profile cli and variable id, page 22
$*\vert^{\pi}_{cli}$	Context with domain π and profile cli, page 22
$\doteq$	Content equivalence of context personal items (cf. Chapters 3, 4), page 24
$\mathcal{A}$	Set of actors in the information system, page 25
$V = (\mathsf{O}, \leftrightarrow)/V_a = (\mathsf{O}_a, \leftrightarrow_a)/V_A = (\mathsf{O}_A, \leftrightarrow_A)$	View on PI Model (of actor $a \in \mathcal{A}$/coalition $A \subset \mathcal{A}$), consisting of set $\mathsf{O} \subset \mathsf{O}^{\mathsf{ctx}}$ of detectable context personal items and equivalence relation $\leftrightarrow$ of associable contexts, page 25
$A \vDash O'$	Record detectability of record $O' \subset \mathcal{O}^{ioi}$ by coalition $A \subset \mathcal{A}$, page 28

Deductive Reasoning (Chapter 3)

$(\mathsf{P}^{\mathsf{ctx}}, \mathcal{P}^{\mathsf{inf}}, \mathbb{P}^{\mathsf{cnt}}, \leftrightarrows, \sigma, \tau)$	Information Model, consisting of context items $\mathsf{P}^{\mathsf{ctx}}$, information items $\mathcal{P}^{\mathsf{inf}}$, contents items $\mathbb{P}^{\mathsf{cnt}}$, related relation $\leftrightarrows$, context-to-information map σ, and information-to-contents map τ (cf. Personal Information Model, Chapter 2), page 35
$\mathsf{P}^{\mathsf{ctx}} := \mathsf{G}^{\mathsf{ctx}} \cup \mathsf{D}^{\mathsf{ctx}} \cup \mathsf{I}^{\mathsf{ctx}}$	Context items (context layer), partitioned into context non-personal items $\mathsf{G}^{\mathsf{ctx}}$, context data items $\mathsf{D}^{\mathsf{ctx}}$, and context identifiers $\mathsf{I}^{\mathsf{ctx}}$, page 35
$\mathcal{P}^{\mathsf{inf}} := \mathcal{G}^{\mathsf{inf}} \cup \mathcal{D}^{\mathsf{inf}} \cup \mathcal{I}^{\mathsf{inf}}$	Information items (information layer), partitioned into non-personal items $\mathcal{G}^{\mathsf{inf}}$, data items $\mathcal{D}^{\mathsf{inf}}$, and identifiers $\mathcal{I}^{\mathsf{inf}}$, page 35
Σ	Signature consisting of function symbols f/k with arity k, page 36
$\mathcal{L}^{ctx}$	Context messages built from context items using function symbols in Σ, page 36
$\mathcal{L}^{x}$	Variable messages built from variables using function symbols in Σ, page 37
$\mathsf{m}@z$	Submessage of context message m at position z ($z = \epsilon$ denotes empty position), page 36

$\doteq$	Content equivalence of context messages (cf. Chapters 2, 4), page 36
$\mathsf{f}(a_1,\ldots,a_k)\leftarrow b_1,\ldots,b_l$	Construction rule for cryptographic primitive, page 37
$\mathsf{f}(\mathsf{m}_1,\ldots,\mathsf{m}_k)\leftarrow \mathsf{n}_1,\ldots,\mathsf{n}_l$	Instantiation of a construction rule, page 37
$\mathsf{f}(a_1,\ldots,a_k)\xrightarrow{\doteq b_1,\ldots,\doteq b_l} c$	Elimination rule for cryptographic primitive, page 38
$\mathsf{f}(\mathsf{m}_1,\ldots,\mathsf{m}_k)\xrightarrow{\mathsf{n}_1,\ldots,\mathsf{n}_l} c$	Instantiation of an elimination rule, page 38
$\mathcal{C}/\mathcal{C}_a/\mathcal{C}_A$	Knowledge base (set of known context messages; of actor $a \in \mathcal{A}$/coalition of actors $A \subset \mathcal{A}$), page 43
$\mathcal{C}\vdash\!\cdots\,\mathsf{m}$	Context message $\mathsf{m} \in \mathcal{L}^{ctx}$ is derivable from set $\mathcal{C} \subset \mathcal{L}^{ctx}$ of context messages, page 40
$\mathcal{C}\vdash\!\cdots^{-}\,\mathsf{m}$	Context message m can be derived from knowledge base $\mathcal{C}$ using elimination rules, page 47
$\mathcal{C}\vdash\!\cdots^{+}\,\mathsf{m}$	Context message m can be derived from knowledge base $\mathcal{C}$ by construction from messages derived using $\vdash\!\cdots^{-}$, page 47
$\mathcal{C}\vdash\!\cdots\,\mathsf{p}_1 \doteq_0 \mathsf{p}_2$	Context items $\mathsf{p}_1, \mathsf{p}_2 \in \mathsf{P}^{\mathrm{ctx}}$ are directly equatable from $\mathcal{C} \subset \mathcal{L}^{ctx}$, page 45
$\mathcal{C}\vdash\!\cdots\,\mathsf{p}_1 \doteq \mathsf{p}_2$	Context items $\mathsf{p}_1, \mathsf{p}_2 \in \mathsf{P}^{\mathrm{ctx}}$ are equatable from $\mathcal{C} \subset \mathcal{L}^{ctx}$, page 45

Equational Reasoning (Chapter 4)

Σ^{eq}	Equational signature consisting of function symbols f/k with arity k (cf. Chapter 3), page 59
$\mathcal{N}$	Set of names representing pieces of information, page 59
$\mathcal{X}$	Set of variables for equational theory, page 60
$\mathcal{T}_{\mathcal{N}}/\mathcal{T}_{\mathcal{X}}/\mathcal{T}_{\mathcal{X}\cup\mathcal{N}}/\mathcal{T}_{\mathsf{P}^{\mathrm{ctx}}}/\mathcal{T}_{\mathbb{P}^{\mathrm{cnt}}}$	"Ground terms" built from names $\mathcal{N}$ using equational signature Σ^{eq}, page 59/"variable terms" built from variables $\mathcal{X}$, page 60/"recipes" built from names and variables, page 61/"context ground terms" built from context items $\mathsf{P}^{\mathrm{ctx}}$, page 67/"contents ground terms" built from contents items $\mathbb{P}^{\mathrm{cnt}}$, page 67
E	Equational theory: finite set of equations $U = V$ where $U, V \in \mathcal{T}_{\mathcal{X}}$, page 60
$=_{\mathsf{E}}/\neq_{\mathsf{E}}$	Equality/inequality of messages under equational theory E, page 60
$\doteq$	Content equivalence of context ground terms (cf. Chapters 2, 3), page 67
$[\mathsf{p}]$	Equivalence class of context item p under content equivalence $\doteq$, page 67
$\tau(\sigma(\mathsf{m}))$	Contents $\in \mathcal{T}_{\mathbb{P}^{\mathrm{cnt}}}$ corresponding to context ground term $\mathsf{m} \in \mathcal{T}_{\mathsf{P}^{\mathrm{ctx}}}$, page 67
$U\sigma/N\phi$	Substitution σ acting on variable term U, page 60/frame ϕ acting on recipe N, page 61
$\phi = \nu n_1,\ldots,n_l.\{m_1/x_1,\ldots,m_k/x_k\}$	Frame representing the knowledge of an actor, page 60
$\phi(\mathcal{C}^{\mathrm{eq}})$	Frame corresponding to equational knowledge base $\mathcal{C}^{\mathrm{eq}}$, page 67
$\phi_{\mathsf{P}}(\mathcal{C}^{\mathrm{eq}})$	Frame corresponding to $\mathcal{C}^{\mathrm{eq}}$ with items in P replaced by a fresh value, page 68
$\phi \vdash M$	Ground term M is deducible from frame ϕ, page 61
$\phi \approx \psi$	Static equivalence of frames ϕ, ψ, page 62
$\mathcal{C}^{\mathrm{eq}}$	Equational knowledge base (set of known context ground terms $\mathcal{T}_{\mathsf{P}^{\mathrm{ctx}}}$, cf. Chapter 3), page 67

$\mathcal{C}^{\mathfrak{c}}/\mathcal{C}^{\mathrm{eq}+\mathfrak{c}}$	Augmented knowledge base $\mathcal{C} \cup \{guess[\cdot]\}$, page 66/equational knowledge base $\mathcal{C}^{\mathrm{eq}} \cup \{guess[\cdot]\}$, where $\tau(\sigma(guess[\cdot])) = \mathfrak{c}$, page 71
$\mathcal{C}^{\mathrm{eq}} \vdash \mathsf{p} \doteq \mathsf{p}'$	Equatability of context items p, p' given equational knowledge base $\mathcal{C}^{\mathrm{eq}}$, page 69

Symbolic Reasoning (Chapter 5)

$\mathfrak{P}^{\mathrm{sym}} := \mathfrak{G}^{\mathrm{sym}} \cup \mathfrak{D}^{\mathrm{sym}} \cup \mathfrak{I}^{\mathrm{sym}}$	Symbolic Information model consisting of symbolic items, partitioned into symbolic non-personal items $\mathfrak{G}^{\mathrm{sym}}$, symbolic data items $\mathfrak{D}^{\mathrm{sym}}$, and symbolic identifiers $\mathfrak{I}^{\mathrm{sym}}$, page 87
$\mathcal{L}^{sym}$	Symbolic messages built from symbolic items using function symbols, page 89
$P/P(r)/P(r)\vert_p$	Symbolic protocol, page 89/Symbolic protocol role, page 89/Symbolic profile, page 97
$v\vert_k/\mathbf{v}\vert_k/\overline{\mathbf{v}}\vert_k$	Non-random/random/instance-random symbolic item $\in \mathfrak{P}^{\mathrm{sym}}$, page 87
$\{\mathfrak{m}\}_F/\mathfrak{m}@_F j/v@_F j\vert_v$	Var-list/var-list element/var-item, page 103
$\mathfrak{m}@_F ?/d@_{all}?\vert_u$	Pattern message/pattern constraint, page 105
$\mathfrak{m}\vert^{\pi}/\mathfrak{C}\vert^{\pi}$	Instantiation of symbolic message $\mathfrak{m}$/set of symbolic messages $\mathfrak{C}$ in domain π, page 89
$\mathfrak{f}(\mathfrak{a}_1, \ldots, \mathfrak{a}_k) \leftarrow \mathfrak{b}_1, \ldots, \mathfrak{b}_l$	Symbolic instantiation of a construction rule, page 92
$\mathfrak{f}(\mathfrak{a}_1, \ldots, \mathfrak{a}_k) \xrightarrow{\doteq \mathfrak{b}_1, \ldots, \doteq \mathfrak{b}_l} \mathfrak{c}$	Symbolic instantiation of an elimination rule, page 92
$\mathbf{T}/\mathbf{F}/\mathfrak{m}/\mathfrak{m} \doteq \mathfrak{m}'/\gamma_1 \wedge \gamma_2/\gamma_1 \vee \gamma_2$	Constraints, page 90
$(\gamma)_l/(\gamma)_r/(\mathfrak{m})_l \doteq (\mathfrak{m}')_r/\gamma_1 \wedge \gamma_2/\gamma_1 \vee \gamma_2$	Biconstraints, page 94
$\mathfrak{C} \vdash^{s} \gamma \Rightarrow \mathfrak{m}$	$\mathfrak{m}$ is symbolically derivable from $\mathfrak{C}$ using constraint γ, page 93
$\mathfrak{C} \vdash^{c} \gamma \Rightarrow \mathfrak{m}$	$\mathfrak{m}$ can be constructed from symbolically derivable messages using constraint γ, page 93
$\gamma \Rightarrow d\vert_p/\gamma \Rightarrow d\vert_p/\boxed{\mathrm{n}}@z$	Profile node/edge/content equivalence node in constraint graph, page 98

Multiple data subjects extension (Section 6.1)

$\mathsf{I}_l^{\mathrm{ctx}}/\mathsf{I}_g^{\mathrm{ctx}}/\mathcal{I}_l^{\mathrm{inf}}/\mathcal{I}_g^{\mathrm{inf}}$	Local/global (context) identifiers, page 108
D/k, I/k, $I/k/l$	Type of personal information: data item/global identifier/local identifier, page 108
$v\vert_{A;B}^{\kappa}$	Context personal item with variable v, domain κ, topic A, and scope B, page 108
$p\vert_i$	ith data subject of personal item $p \in \mathcal{O}^{\mathrm{inf}}$, page 108

Attribute predicates extension (Section 6.2)

$d?_p\vert_k^{\pi} \in \mathsf{R}^{\mathrm{ctx}}$	Context predicate item, page 111
$p?_{pr} \in \mathcal{R}^{\mathrm{inf}}$	Predicate item, page 111

States, traces, and system evolution (Section 6.3)

$\{\mathcal{C}_x\}_{x \in \mathcal{A}}$	State: collection of knowledge bases $\mathcal{C}_x$ of actors $x \in \mathcal{A}$, page 114

$a(\mathsf{id}_a) \to b(\mathsf{id}_b) \ : \ \mathsf{m}$	Transmission from actor a (with communication address id_a) to actor b (with communication address id_b) of message m, page 114
$\mathfrak{T} = \mathsf{t}_1; \cdots; \mathsf{t}_k$	Trace: sequence of transmissions, page 114

Models of cryptographic primitives (for the rule-based model)

$\mathsf{aencl}(m, l, \mathsf{pk}(k^-))$	Labelled asymmetric encryption of message m with label l using public key $\mathsf{pk}(k^-)$, page 142
$\mathsf{aenc}(m, \mathsf{pk}(k))$	Asymmetric encryption of message m using public key $\mathsf{pk}(k)$, page 42
$\mathsf{aka}(k_1^-, n_1, k_2^-, n_2)$	Session key derived from private keys k_1^-, k_2^- and randomness n_1, n_2 using authenticated key agreement, page 142
$\mathsf{cred}(i, k^-, d, n_o, n_i)$	Anonymous credential with owner identifier i; issuer secret key k^-, owner attributes d, owner-contributed randomness n_o, and issuer-contributed randomness n_i, page 122
$\mathsf{enc}(m, k)$	Symmetric encryption of message m under key k, page 42
$\mathsf{h}(x)$	Cryptographic hash of message x, page 42
$\mathsf{icred}(i, k^-, d, n_1, n_o, n_3, n_4, n_i, n_6, n_7)$	Issuing protocol of an anonymous credential $\mathsf{cred}(i, k^-, \mathsf{d}, n_o, n_i)$, in which the owner additionally contributes randomness n_1, n_3, n_7, and the issuer additionally contributes randomness n_4, n_6, page 122
$\mathsf{pc}(s, b, d)$	Parelsnoer pseudocode based on BSN b and domain d using secret s, page 164
$\mathsf{pk}(k)$	Public key corresponding to private key k, page 42
$\mathsf{rc}(x, y)$	Commitment to x using randomness y, page 122
$\mathsf{sig}(m, k)$	Digital signature (with message recovery) on message m signed with private key k, page 42
$\mathsf{zk}(s, p, n_1, n_2)$	Zero-knowledge proof of knowledge with secret s, public information p, randomness n_1 for the commitment of the prover, and challenge n_2 of the verifier, page 118
$\{m_1, \ldots, m_k\}$	Concatenation of messages $m_1, \ldots, m_k$, page 41

Index