Noname manuscript No.
(will be inserted by the editor)

Data Minimisation in Communication Protocols: A Formal Analysis Framework and Application to Identity Management

Meilof Veeningen · Benne de Weger · Nicola Zannone

the date of receipt and acceptance should be inserted later

Abstract With the growing amount of personal information exchanged over the Internet, privacy is becoming more and more a concern for users. One of the key principles in protecting privacy is data minimisation. This principle requires that only the minimum amount of information necessary to accomplish a certain goal is collected and processed. "Privacy-enhancing" communication protocols have been proposed to guarantee data minimisation in a wide range of applications. However, currently there is no satisfactory way to assess and compare the privacy they offer in a precise way: existing analyses are either too informal and high-level, or specific for one particular system. In this work, we propose a general formal framework to analyse and compare communication protocols with respect to privacy by data minimisation. Privacy requirements are formalised independent of a particular protocol in terms of the knowledge of (coalitions of) actors in a three-layer model of personal information. These requirements are then verified automatically for particular protocols by computing this knowledge from a description of their communication. We validate our framework in an identity management (IdM) case study. As IdM systems are used more and more to satisfy the increasing need for reliable on-line identification and authentication, privacy is becoming an increasingly critical issue. We use our framework to analyse and compare four identity management systems. Finally, we discuss the completeness and (re)usability of the proposed framework.

Keywords Privacy, Identity Management, Formal methods, Data minimisation, Detectability, Associability

Meilof Veeningen (✉) · Benne de Weger · Nicola Zannone
Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands

Meilof Veeningen
E-mail: m.veeningen@tue.nl

1 Introduction

As more and more personal information is exchanged over the Internet by businesses and other organisations, privacy risks are becoming a major concern. For instance, e-health and identity management systems deal with large amounts of personal information. There have been numerous reports of information from such systems being used for secondary purposes [94], or being stolen and abused by third parties [80]. 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 means limiting the amount of shared personal information, but also limiting the use of identifiers that different actors can use to correlate their views on a data subject [53].

One important approach to achieve privacy by data minimisation is the use of *privacy-enhancing* communication protocols [53,93]. 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 [82], e-voting [50], and electronic toll collection [42].

Understanding the privacy differences between privacy-enhancing protocols designed for the same purpose is important, e.g., for system designers who want to use privacy-enhancing protocols, or for system administrators who want to select what system to use. However, it is typically not straightforward to obtain such an understanding. One reason is that privacy-enhancing protocols typically combine (advanced) cryptographic primitives in subtle ways; also, typical scenarios involve multiple actors which may collude

in different coalitions to correlate their views on data subjects. Existing comparisons in areas such as e-health [74,92] or identity management [1,55] are performed in an informal and high-level (and thus, possibly subjective) way, and thus their privacy assessments do not offer much insight into differences between systems and the reasons behind them. On the other hand, proposals for privacy-enhancing systems typically assess the privacy of their own solution using terminology and criteria specific to the setting at hand, making it hard to compare different systems. Thus, we need a practical way to compare different systems that is precise and verifiable, yet application-independent; and that provides sufficient detail for real insight into the privacy differences that exist between systems.

Formal methods provide the machinery to perform such a comparison. Over the years, formal methods, e.g., the applied pi calculus [2], have arisen as an important tool to analyse security of communication in IT systems [2,22,68, 78]. The idea is to express communication protocols in a suitable formalism, and then verify whether such a model of the protocol satisfies, e.g., authentication properties [22] or secrecy properties [10]. Secrecy, in particular, expresses one aspect of privacy; namely, whether a certain piece of information is known by some party in a protocol. However, it leaves unanswered a question which is equally important for privacy-enhancing protocols; namely, whether a certain piece of information can be linked to its corresponding data subject (who, in general, might not be a direct participant in the communication under analysis).

Recently, several research efforts have focused on the analysis of privacy properties using the applied pi calculus and related techniques [21,42,43,45,91], in application domains such as electronic toll collection [42], e-voting [43, 45], and RFID systems [21]. While this approach has delivered considerable successes, several issues inhibit its use for our purposes, namely, practical and accessible high-level privacy analysis. First, in many cases, properties are defined and verified specific to their respective settings or protocols [42,43,45]. General definitions for the common privacy property of linkability exist [5], but they are aimed towards linking messages to their senders (whereas data minimisation concerns linking of information to its data subject) and defined with respect to an outside attacker (whereas data minimisation concerns the knowledge of actors or coalitions inside the system). Second, such methods require considerable manual work for each property to be verified, in many cases including particular assumptions on the model to make computation feasible. Third, analysis results are not summarised in a comprehensive and intuitive way, necessitating substantial manual review.

In our previous works [96,99], we have introduced building blocks for high-level privacy analysis of protocols to exchange personal information. We introduced a three-layer model that captures the knowledge of personal information held by different (coalitions of) communicating parties [96, 99]. The model captures the context in which pieces of information has been observed, as well as the contents they have. We showed how relevant privacy requirements can be expressed as properties of items in this model. We also showed how this model is determined from observations of communication between the different parties. However, the model of [96,99] only captures communication that uses a limited set of cryptographic primitives; moreover, it does not offer an implementation of the analysis method; and finally, it does not discuss in general what kinds of privacy requirements can be verified, or how to perform a privacy comparison in practice.

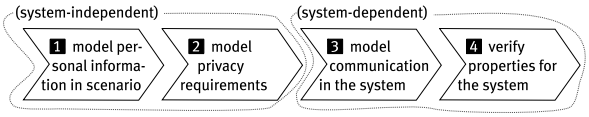


Fig. 1 Steps of our privacy comparison framework

In this work, we combine our previous building blocks into a general framework for privacy comparison of communication protocols, and we apply the framework in an identity management case study. Specifically, our contributions are as follows:

- We present a framework to compare communication protocols with respect to privacy by data minimisation. Our framework gives precise, verifiable results with enough detail to obtain insight into privacy differences;
- We extend our previous formal method [96,99] for the analysis of knowledge of personal information to cover additional primitives and cryptographic protocols (specifically, zero-knowledge proofs and issuing protocols for anonymous credentials);
- We provide an implementation of the formal method in Prolog to automate part of the comparison;
- We validate our framework by analysing and comparing four identity management systems: we show that a range of relevant privacy requirements can be captured by our model, and use our framework to formally analyse the identity management systems with respect to these requirements.

Our privacy comparison framework consists of four steps, shown in Figure 1. The first two steps are to model the scenario and its requirements. We introduce two formalisms: the *Personal Information (PI) Model* (§2.1) to model different types of personal information and their relations; and the *view* of an actor to describe the partial knowledge about this information that this actor has at one point in time (§2.2). The *first step* of our method comprises modelling all personal information using a PI model, and modelling the initial knowledge of each actor as a view on that PI model. This means modelling the personal information as used in

the protocol instances in the scenario; however, it also means modelling other knowledge of personal information held initially by the actors. This way, we can assess how links can be established between the knowledge learned from the protocol instances and the initial knowledge. The *second step* is to model data minimisation requirements, i.e., which personal information should become known or remain unknown to which actors in the system. These requirements are phrased as properties 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 then to model the exchange of information in the information systems. For this, we need to model the evolution of actor knowledge in such systems due to the exchange of messages. We extend the PI model into an *information model* that also includes messages using cryptographic primitives, and the non-personal information they may contain. We express the messages that an actor has exchanged at a certain point in time using the notion of a *knowledge base* on that information model (§3.1). We define a procedure to determine an actor's view from his knowledge base (§3.2), and present an algorithm that implements it (§3.3). Finally, we introduce states to formalise the knowledge of all actors in the system at one point in time, and traces to capture a series of communications that transforms one state into another (§4).

The *fourth step* is to verify which systems satisfy which requirements. This step is performed automatically using our Prolog implementation[1]. Given a PI model, set of formalised requirements, initial state and trace, this tool first determines the state of the system after communication; then uses our formal procedure to compute the corresponding views of the actors in the system, and finally determines which requirements hold in these views.

We validate our framework by applying it to an identity management case study. Identity management (IdM) systems [89,58,47] offer reliable on-line identification and authentication 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 [8, 31,100]. However, while high-level sketches of privacy issues [3,12,53,61] and comparisons of systems [1,55] exist, no comprehensive set of relevant privacy requirements 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. In Section 5, we present our case study: we introduce IdM, discuss privacy requirements for IdM systems, and introduce four IdM systems. In Section 6, we use our framework to formally compare the privacy offered by these four IdM systems with respect to the requirements introduced above, and discuss the results.

Finally, we discuss the completeness and (re)usability of our framework (§7). We conclude the paper by discussing related work (§8), drawing conclusions, and pointing to interesting directions for future work (§9).

[1] The tool and formal model of our case study are available at `www.mobiman.me/downloads/`.

2 A Model for Knowledge of Personal Information

In this section, we present the Personal Information (PI) Model and actor views. A *PI model* (§2.1) describes personal information in an information system at a certain point in time; the *view* of an actor involved in the system on this PI model (§2.2) captures the knowledge about this information held by that actor. Privacy requirements (§2.3) are modelled as properties of items from these views. The PI model is used in step 1 of our framework (Figure 1) to model personal information, and it is the basis for the model of communication in step 3. Views are used in step 1 to express initial knowledge of actors; in step 2 to model requirements; and in step 4 to compare actual knowledge to these requirements. Our model is based on two main assumptions:

- Discrete information — There is a finite set of pieces of personal information that each belong to a particular data subject. In particular, we allow knowledge about finitely many boolean predicates on personal information (e.g., "Alice's age is below k" for some particular value k). 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 [68], and simplify both the specification of requirements and the modelling of protocols. We discuss approaches that do not make these abstractions in Section 8.

2.1 Personal Information Model

A Personal Information (PI) model describes all personal information present in an information system at a certain point in time.

2.1.1 Personal Information

A piece of personal information in the PI model is a *specific* string that has a *specific* meaning as personal information about a *specific* person, e.g., "the age of Alice". We distinguish between two types of digital personal information: *identifiers* and *data items*. Identifiers are unique within the system; for data items, this is not necessarily the case. The sets of identifiers and data items are denoted $\mathcal{I}$ and $\mathcal{D}$, respectively. We express that pieces of personal information satisfy certain fixed boolean *properties* relevant to the application domain by a set $\{\psi_1, \ldots, \psi_k\}$ of partial functions $\mathcal{I} \cup \mathcal{D} \to \mathcal{D}$ that assign properties to some of the identifiers and data items. For instance, suppose ψ_j represents the property that an age is over 60. If $\psi_j(d)$ is defined, i.e., d has an image under partial function ψ_j, then d represents the age of a data subject who is over 60 and $\psi_j(d)$ represents the fact that this data subject has an age over 60. If $\psi_j(d)$ is not defined, i.e., d does not have an image under partial function ψ_j, then either d does not represent an age, or it represents an age below 60. The set $\mathcal{E}$ of *entities* models the real-world persons whom the considered information is about. Elements of the set $\mathcal{O} := \mathcal{E} \cup \mathcal{I} \cup \mathcal{D}$ are called *items of interest*. The link between information and its subject is captured by the *related* relation, an equivalence relation on $\mathcal{O}$ denoted $\Leftrightarrow$. Namely, given two items of interest $o_1, o_2 \in \mathcal{O}$, $o_1 \Leftrightarrow o_2$ means that o_1 and o_2 are information about the same person.

These concepts, however, are 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 know how an actor obtained a piece of information. We assumed that actors combine pieces of information that they know belong to the same data subject. However, if an actor learns the same piece of information (e.g., "the age of Alice") several times (e.g., in two protocol instances with different session identifiers), he may not know that it is the same information. Thus, to represent the knowledge of this actor, we need to distinguish between these two "representations" of the information.

2.1.2 Three-Layer Model

Because of the need to distinguish different instances of the same piece of information, as well as to reason about message contents, we introduce a three-layer representation of personal information. The representation consists of the *context layer*, *information layer*, and *contents layer*. At the information layer, as described above, the information itself is represented, e.g., "Alice's city". At the context layer, information is described in terms of the context in which it has been observed, e.g., "the city of the user in protocol instance #1". At the contents layer, information is described in terms of the strings actually transmitted in a protocol, e.g., "Eindhoven". Actor knowledge is modelled using the context layer and reasoned about using the contents layer; the information layer is used to specify privacy requirements or visualise analysis results [97].

At the context layer, we model the *context* in which an actor knows pieces of information. A context is an item $*|_k^\eta$, where η is called the *domain*, and k is called the *profile*. A domain is any separate digital "place" where personal information is stored or transmitted, e.g., a database or an instance of a communication protocol. A profile represents a particular data subject in a domain, e.g., an entry about one person in a database, or a logical role in a protocol instance (however, different profiles in a domain may still represent the same data subject, e.g., duplicate entries in a database).

In such a context, pieces of information are represented by *variables*. A variable describes the type of information in the context of that domain, e.g. "session identifier" or "age attribute". Namely, the piece of information with variable v in context $*|_k^\eta$ is denoted $v|_k^\eta$. Context-layer representations of entities, identifiers, and data items are modelled by *context entities* E^c, *context identities* I^c, and *context data items* D^c, respectively. The set $\mathsf{O}^c := \mathsf{E}^c \cup \mathsf{I}^c \cup \mathsf{D}^c$ is the set of all *context personal items*. The unique context entity in context $*|_k^\eta$ is denoted $ds|_k^\eta$. Properties of identifiers and data items are modelled at the context layer by extending the partial functions ψ_i above.

We represent personal information at the contents layer as elements from an arbitrary set $\mathfrak{C}$ of *message contents*. 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.

Apart from these three descriptions of pieces of personal information, the PI model also defines mappings between the three 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:

Definition 1 A *Personal Information (PI) Model* is a tuple

$$(\mathsf{O}^c, \mathcal{O}, \Leftrightarrow, \sigma, \tau, \{\psi_1, \ldots, \psi_k\})$$

so that:

- O^c is a set of *context personal items*, partitioned into $\mathsf{O}^c = \mathsf{E}^c \cup \mathsf{I}^c \cup \mathsf{D}^c$. Here, E^c are *context entities* $ds|_n^\kappa$ with

arbitrary domain κ and profile n; I^c and D^c are *context identifiers* and *context data items* $v|_n^\kappa$ with arbitrary variable v, domain κ and profile n, respectively;
- $\mathcal{O}$ is a set of *items of interest*, partitioned into $\mathcal{O} = \mathcal{E} \cup \mathcal{I} \cup \mathcal{D}$. Here, $\mathcal{E}$ are *entities*; $\mathcal{I}$ are *identifiers*; and $\mathcal{D}$ are *data items*;
- $\Leftrightarrow \subset \mathcal{O} \times \mathcal{O}$ is the *related* relation on $\mathcal{O}$: an equivalence relation so that every item of interest is related to exactly one entity;
- σ is a map $\mathsf{O}^c \to \mathcal{O}$ so that $\sigma(\mathsf{E}^c) \subset \mathcal{E}$; $\sigma(\mathsf{I}^c) \subset \mathcal{I}$; and $\sigma(\mathsf{D}^c) \subset \mathcal{D}$; and $\sigma(x|_k^\eta) \Leftrightarrow \sigma(y|_k^\eta)$ for any $x|_k^\eta, y|_k^\eta$;
- τ is a map $\mathcal{I} \cup \mathcal{D} \to \mathfrak{C}$ so that $\forall i_1, i_2 \in \mathcal{I}$: if $\tau(i_1) = \tau(i_2)$, then $i_1 = i_2$;
- $\{\psi_1, \ldots, \psi_n\}$ are partial functions $\psi_i : \mathsf{I}^c \cup \mathsf{D}^c \to \mathsf{D}^c$, $\mathcal{I} \cup \mathcal{D} \to \mathcal{D}$ so that: 1) $\psi_i(o)$ is related to $o \in \mathcal{I} \cup \mathcal{D}$ whenever defined; 2) $\tau(\psi_i(o_1)) = \tau(\psi_i(o_2))$ whenever defined for some $o_1, o_2 \in \mathcal{I} \cup \mathcal{D}$; 3) $\psi_i(\mathsf{o})$ has the same context as $\mathsf{o} \in \mathsf{I}^c \cup \mathsf{D}^c$ whenever defined; 4) $\sigma(\psi_i(\mathsf{o})) = \psi_i(\sigma(\mathsf{o}))$ for every $\mathsf{o} \in \mathsf{I}^c \cup \mathsf{D}^c$ for which $\psi_i(\mathsf{o})$ is defined.

The first two bullets of the definition define information at the context and information layers, respectively; the third bullet defines personal relations at the information layer. The fourth and fifth bullet define the mapping between the three layers: we demand that the contents of identifiers are unique. The sixth bullet introduces properties both at the context and information layers. Properties at the information layer preserve relation $\Leftrightarrow$ (1) and have contents independent from the item they are a property of (2); properties at the context layer preserve context (3) and are consistent with the information-layer properties (4).

We introduce notation for context personal items $x|_k^\eta, y|_l^\chi$ representing the same information or contents. If $\sigma(x|_k^\eta) = \sigma(y|_l^\chi)$, then we write $x|_k^\eta \equiv y|_l^\chi$ and we call $x|_k^\eta$ and $y|_l^\chi$ *equivalent*. 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*. Clearly, equivalence implies content equivalence.

The next example shows a PI model as used in step 1 of our analysis framework, i.e., to model all personal information present in a particular scenario.

Example 1 Figure 2 shows a PI model representing personal information about two entities, Alice ($al \in \mathcal{E}$) and Bob ($bob \in \mathcal{E}$), in a simple scenario. Recall that a PI model is used to express all personal information in a scenario, regardless of which protocols are used; regardless of who knows the information, and also including other information that it may be linked to by the actors involved. In this scenario, a client and a server exchange information about Alice. Namely, the server has a database with personal information about different entities; the server and client engage in a protocol to exchange information about Alice; and the client combines the results with her address book.

At the information layer of this PI model, Alice has identifier id_a and an age age_a; Bob has identifier id_b, age age_b, and telephone number $teln_b$. Alice and Bob happen to have the same age, so $\tau(age_a) = \tau(age_b)$; the other pieces of information have distinct contents. (This example does not consider attribute properties.)

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. The data subject of profile k is represented by context entity $ds|_k^{db}$.
- 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 an identifier $id|_4^{ab}$; the 12th entry contains a telephone number $teln|_{12}^{ab}$.
- 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.

(In a full analysis using our framework, we would also model the client and server as entities. This allows us to reason about knowledge about their involvement in this scenario. For simplicity, we omit them here.) □

2.2 Views: Actor Knowledge

Each actor in an information system has partial knowledge about the personal information in that system. Our framework analyses privacy by data minimisation by verifying that this partial knowledge satisfies certain requirements. We model actors as a finite set $\mathcal{A}$. We require each actor to be an entity in the PI model, i.e., $\mathcal{A} \subset \mathcal{E}$. The knowledge of an actor at some point in time consists of knowledge of some pieces of personal information, and knowledge that some of these pieces of information are about the same person. We capture this knowledge as a *view* on the PI model, consisting of a set of context-layer items and an equivalence relation on them:

Definition 2 Let $M = (\mathsf{O}^c, \mathcal{O}, \Leftrightarrow, \sigma, \tau, \{\psi_1, \ldots, \psi_k\})$ be a PI Model. A *view* on M is a tuple $V = (\mathsf{O}_*, \leftrightarrow_*)$ such that:
- $\mathsf{O}_* \subset \mathsf{O}^c$ is the set of *detectable* items;
- $\leftrightarrow_* \subset \mathsf{O}_* \times \mathsf{O}_*$ is the *associability* relation: an equivalence relation on O_*.

The view of actor $a \in \mathcal{A}$, determined in step 4 of our framework, is denoted $V_a = (\mathsf{O}_a, \leftrightarrow_a)$. From a privacy perspective, we are not just interested in the views of single actors

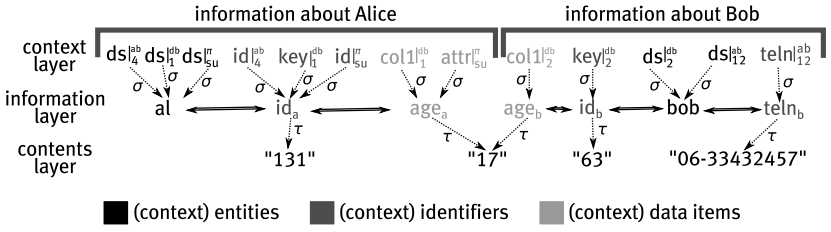


Fig. 2 Personal Information Model of Example 1

$a \in \mathcal{A}$, but also in the views of coalitions $A \subset \mathcal{A}$. Such a view represents knowledge of personal information when the actors in the coalition combine all information (e.g., databases, communication transcripts) they have. The view of coalition $A \subset \mathcal{A}$ after communication is denoted $V_A = (\mathsf{O}_A, \leftrightarrow_A)$. It contains at least the knowledge of each individual actor in the coalition.

We next show an example of the views of actors and coalitions, as they may be obtained in step four of our framework when analysing a particular communication protocol.

Example 2 Consider the PI model M from Example 1. The actors in this information system are the client and server, i.e., $\mathcal{A} = \{c, s\}$. Figure 3 shows example views of these actors 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. Namely, the client knows Bob's telephone number $teln_{bob}$ as entry $teln|^{ab}_{12} \in \mathsf{O}_c$ in her telephone book; she also knows that this is Bob's telephone number, expressed by detectability $ds|^{ab}_{12} \in \mathsf{O}_c$ and associability $ds|^{ab}_{12} \leftrightarrow_c teln|^{ab}_{12}$. About Alice, the client knows two context-layer representations of identifier id_a: as part of her telephone book entry ($id|^{ab}_4 \in \mathsf{O}^c$), and as a piece of information sent in protocol instance π ($id|^{\pi}_{su} \in \mathsf{O}^c$). She again knows the data subject corresponding to the telephone book entry ($ds|^{ab}_4$), and she knows the age transmitted in the protocol ($attr|^{\pi}_{su} \in \mathsf{O}^c$). Moreover, she can associate the information in the address book to the information from the protocol instance.

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 mutually associable pieces of information $col1|^{db}_2$, $key|^{db}_2$ from the database. About Alice, the server also knows two associable pieces of information 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.

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 because both actors know the same identifier (in the figure, all context personal items about Alice are connected by arrows). However, information about Bob is divided into two equivalence classes: the client knows entity *bob* and his telephone number $teln_{bob}$ and the server knows age age_b and telephone number $teln_b$, but they cannot associate this information to each other (indicated by the absence of arrows between the information in the figure). □

2.3 Privacy Requirements

The second step of our analysis framework is to model each relevant data minimisation requirement in terms of the views of actors and coalitions. This includes both modelling functional requirements, i.e., modelling what *should* be learned by the actors in the protocol, and modelling privacy requirements, i.e., modelling what *should not* be learned. These requirements are formulated independently from any particular system, then verified for each particular system modelled. Thus, our framework can be used to generically verify any requirement that can be phrased in terms of views, including:

- *Detectability requirements* — Can a given actor/coalition of actors detect a given piece of information, or a given context-layer representation?
- *Linkability requirements* — Can a given actor/coalition of actors associate given contexts, or any contexts in which he knows given pieces of information?
- *Involvement requirements* — Is there a domain d in which an actor can associate one profile to a given context c_1, and another 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 requirements can be defined as arbitrary combinations of these elementary requirements and their negations. The next example shows different types of requirements.

Example 3 We formulate three requirements for the scenario given in Example 1. Recall that we have actors $\mathcal{A} = \{c, s\}$ with views

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

First, since the goal of the protocol is to exchange information, we can check whether the client has indeed learned the

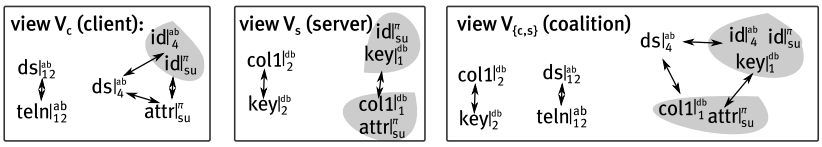


Fig. 3 Views of actors c and s and coalition $\{c,s\}$ in a scenario (Example 2). Detectable context personal items are shown; grey areas are sets of equivalent items. Associations are represented by arrows; for simplicity, they are shown up to equivalence.

age of 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 requirement and a linkability requirement, 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 requirement).

Now consider the views in the particular system from Example 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 3, left), while in view $V_{\{c,s\}}$, $teln|^{ab}_{12} \leftrightarrow_{c,s} col1|^{db}_2$ is not true (Figure 3, right). □

We show additional examples of the different types requirements in Section 6 when analysing identity management systems. In Section 7, we discuss what kind of requirements cannot be represented in this way.

3 Deducing Views from Communicated Messages

In this section, we determine the views of actors by modelling and analysing the messages they have exchanged. We present the *information model*, capturing messages containing personal information; and *knowledge bases*, capturing which messages an actor has observed at a certain point in time. We then propose a formal procedure to derive an actor's view from his knowledge base. This procedure is based on the following main assumptions:

- *Detecting from messages* — We model messages using a Dolev-Yao-style black-box model of cryptography. A piece of personal information is detected using a message it occurs in by: 1) reading it from that message; 2) applying a cryptographic operation on the message that uses the information; or 3) comparing the message's contents to another message whose structure is known.
- *Associating by identifiers* — Contexts are associated to each other by observing that the same identifier or entity occurs in both contexts.

The modelling of cryptographic primitives and operations as "terms" in a black-box model is standard ever since the seminal work by Dolev and Yao [44]. Determining what knowledge can be read from such messages can be done using standard deductive systems [37,44,49]. When adapting these standard techniques to our model of personal information, observing the application of cryptographic operations and comparing the contents of messages are needed as extensions. These three ways of deriving personal information also occur in the popular equational approach using static equivalences [16]; see Section 8 for a comparison.

Defining associability by identifiers is suitable for our goal, namely, comparing different protocols with respect to the knowledge that actors learn. Namely, protocols differ in what identifiers they use and how; our definition of associability allows us to reason about the privacy consequences this has. Associability does not take into account probabilistic links due to (combinations of) non-identifying personal information; probabilistic linking methods are orthogonal to our approach (see Section 8).

The formalisation of messages and knowledge bases is described in Section 3.1; the methodology for determining views from knowledges base is described in Section 3.2.

3.1 Messages, Information Model, Knowledge Base

Communication in privacy-enhancing protocols uses messages built up from personal and other information, e.g., nonces and session keys. At the context layer of our three-layer model, non-personal information is modelled by a set G^c of *context non-personal items*. Items in G^c belong to a domain, but not to a profile: in this case we denote the profile as $\cdot$, e.g. $shakey|^{\eta}_{\cdot}$. At the information layer, we define set $\mathcal{G}$ of *non-personal items*.

Messages built from personal and non-personal information using cryptographic primitives such as encryption, signatures, and hashes, are defined using a grammar. Figure 1 shows the grammar for the primitives used to model the identity management architectures presented in Section 5. For instance, $S_*(*)$ represents digital signatures: if $k^-|^{\pi}_s \in \mathsf{I}^c$ is a private key and $d|^{\pi}_{su} \in \mathsf{D}^c$ is a data item, then $S_{k^-|^{\pi}_s}(d|^{\pi}_{su})$ is a digital signature on the data item using the key. (In this case, we write $S_{k^-|_s}(d|_{su})|^{\pi}$ as shorthand.) Although we model particular primitives here, our approach in general is independent from the particular primitives that are used; in Section 7, we offer some insight into the effort needed to model other primitives. As usual (e.g., [17]), the public key belonging to private key k^- is represented as $\mathsf{pk}(\mathsf{k}^-)$.

We model the following cryptographic primitives. Concatenation, hashing, and (a)symmetric encryption are modelled as usual [37,49]. Digital signatures are "with appendix" [69]: that is, an actor needs to know the message that was signed in order to verify the signature. Labelled asymmetric encryption [8] is asymmetric encryption to which a label

Messages	Meaning
$M,M_i ::= \varnothing$ |	empty message
p |	information (for $\mathcal{L}^c$: $\mathsf{p} \in \mathsf{I}^c \cup \mathsf{D}^c \cup \mathsf{G}^c$; for $\mathcal{L}$: $\mathsf{p} \in \mathcal{I} \cup \mathcal{D} \cup \mathcal{G}$)
$\mathsf{pk}(M_1)$ |	public key corresponding to private key M_1
$\{M_1,\ldots,M_n\}$ |	concatenation of messages $M_1,\ldots,M_n$
$\mathcal{H}(M_1)$ |	hash of message M
$E'_{M_1}(M_2)$ |	symmetric encryption of message M_2 with key M_1
$E_{M_1}(M_2)$ |	asymmetric encryption of message M_2 with public key M_1
$S_{M_1}(M_2)$ |	digital signature of message M_2 with private key M_1
$E_{M_1}(M_2)_{M_3}$ |	labelled asymmetric encryption of message M_2 with public key M_1 and label M_3
$\mathrm{AKA}(M_1;M_2;M_3;M_4)$ |	derived key from authenticated key agreement (AKA) with (SK,randomness) pairs (M_1,M_2) and (M_3,M_4)
$\mathrm{cred}^{M_1}_{M_2}(M_3;M_4)$ |	anonymous credential with user identifier M_1, issuer private key M_2, attributes M_3, and randomness M_4
$\mathrm{ZK}(M_1;M_2;M_3;M_4)$ |	zero-knowledge proof of knowledge of secret M_1 with properties M_3 using public information M_2 and randomness M_4
$\mathrm{ICred}^{M_1}_{M_2}(M_3;M_4)$	issuing protocol for anonymous credential $\mathrm{cred}^{M_1}_{M_2}(M_3;M'_4)$, where M'_4 is derived from M_4

Table 1 Grammar defining sets of cryptographic messages at context layer ($\mathcal{L}^c$) and information layer

is unmodifiably attached at encryption time. For instance, the label can represent a policy specifying when the recipient is allowed to decrypt the data. Authenticated key agreement (AKA) [63] allows two parties to derive a unique session key based on secret keys and randomness contributed by both parties. We consider the variant presented in [63] in which both parties send each other a random value. Both parties can determine the session key, modelled by the AKA primitive, from one private key, the other public key, and the randomness. The *cred* primitive models anonymous credentials [8]. Message $\mathrm{cred}^{M_1}_{M_2}(M_3;M_4)$ represents an endorsement with private key M_2 that the attributes M_3 belong to the user with identifier M_1, randomised using M_4.

We also model two-party cryptographic protocols. Using these protocols, anonymous credentials can be issued without the issuer obtaining the credential or learning M_1; also, their ownership can be proven without revealing the credential itself. Such protocols only have meaning when looked at as a whole, i.e., the meaning lies not in individual messages, but in their combination in a particular order. Thus, we model the complete transcript (i.e., all messages of all participants) of such a protocol as one grammar element. We introduce two such primitives.

First, we model a family of zero-knowledge (ZK) proofs (e.g., [41]) by means of the ZK primitive. In a ZK proof for a given property, a prover wants to convince a verifier that he knows some secrets satisfying that property with respect to some given public information, without revealing anything about the secrets. Here, we consider ZK proofs proving that (1) the public information has a certain message structure with respect to the private information, and (2) some secret attributes d_i satisfy some properties $\psi_k(\mathsf{d}_i)$. For instance, $\mathrm{ZK}(\{\mathsf{d},\mathsf{n}\};\mathcal{H}(\{\mathsf{d},\mathsf{n}\});\psi_2(\mathsf{d});\mathsf{n}')$ denotes a ZK proof (using randomness n') convincing a verifier knowing the hash $\mathcal{H}(\{\mathsf{d},\mathsf{n}\})$ that the prover knows the pre-image $\{\mathsf{d},\mathsf{n}\}$ of the hash, and that $\psi_2(\mathsf{d})$ is satisfied; without the verifier learning anything else about d or n. See Appendix B.1 for a detailed discussion.

Second, we model the issuing protocol for anonymous credentials [8] by means of the *ICred* primitive. This protocol is run between a user and an issuer. In advance, both parties need to know the attributes to be certified, but only the user needs to know the identifier to which the attributes are issued. As a result of the protocol, the user obtains an anonymous credential linking the attributes to the identifier. The issuer does not learn the credential; moreover, because he does not know the identifier, he cannot issue credentials in her name without her involvement. Also, by using ZK proofs for proving ownership, the credential can be "shown" without revealing the identifier. See Appendix B.2 for details.

Formally, we define an *information model* that extends the personal information model from Definition 1 by adding non-personal information and messages:

Definition 3 An *information model* is a tuple

$$(\mathcal{L}^c,\mathcal{L},\mathsf{E}^c,\mathcal{E},\Leftrightarrow,\sigma,\tau,\{\psi_1,\ldots,\psi_k\})$$

so that:

- The set $\mathcal{L}^c$ of *context messages* consists of sets I^c of *context identifiers*, D^c of *context data items*, and G^c of *context non-personal items*, and messages built from them using the grammar of Table 1. Here, I^c and D^c are as in Definition 1; G^c consists of items $p[\!]^{\eta}$ with arbitrary variable p and domain η; the set $\mathsf{P}^c := \mathsf{I}^c \cup \mathsf{D}^c \cup \mathsf{G}^c$ is the set of *context items*;
- The set $\mathcal{L}$ of *information messages* consists of sets $\mathcal{I}$ of *identifiers* and $\mathcal{D}$ of data items as in Definition 1; $\mathcal{G}$ of non-personal items; and messages built from them using the grammar of Table 1;
- Sets E^c of *context entities* and $\mathcal{E}$ of *entities*, and the *related* relation $\Leftrightarrow$ on $\mathcal{O} = \mathcal{E} \cup \mathcal{I} \cup \mathcal{D}$ are as in Definition 1.

- σ is a map $\mathcal{L}^c \cup \mathsf{E}^c \to \mathcal{L} \cup \mathcal{E}$; as a map $\sigma|_{\mathsf{O}^c} : \mathsf{O}^c \to \mathcal{O}$, σ is as in Definition 1 (where $\mathsf{O}^c = \mathsf{E}^c \cup \mathsf{I}^c \cup \mathsf{D}^c$, and $\mathcal{O} = \mathcal{E} \cup \mathcal{I} \cup \mathcal{D}$); also, $\sigma(\mathsf{G}^c) \subset \mathcal{G}$, and σ preserves the grammar structure of messages;
- τ is a map $\mathcal{L} \to \mathfrak{C}$; $\tau|_{\mathcal{I} \cup \mathcal{D}}$ is as in Definition 1; $\tau(E'_x(y)) = \tau(z)$ iff $z = E'_{x'}(y')$, $\tau(x) = \tau(x')$ and $\tau(y) = \tau(y')$; and similarly for other primitives.
- $\{\psi_1, \ldots, \psi_k\}$ are partial functions $\psi_i : \mathsf{I}^c \cup \mathsf{D}^c \to \mathsf{D}^c$, $\mathcal{I} \cup \mathcal{D} \to \mathcal{D}$ as in Definition 1.

In particular, $(\mathsf{O}^c, \mathcal{O}, \Leftrightarrow, \sigma, \tau, \{\psi_1, \ldots, \psi_k\})$ in the above definition is a PI model. Note that (context) entities cannot occur in messages, so we mention them explicitly in the tuple defining an information model. The maps σ and τ preserve grammar structure; for instance, if $\sigma(\mathsf{k}^-) = sk_{al}$ and $\sigma(\mathsf{d}) = age_{bob}$, then $\sigma(S_{\mathsf{k}^-}(\mathsf{d})) = S_{sk_{al}}(age_{bob})$. Like pieces of personal information, we call context messages m and n in general *equivalent* iff $\sigma(\mathsf{m}) = \sigma(\mathsf{n})$, and *content equivalent* iff $\tau(\sigma(\mathsf{m})) = \tau(\sigma(\mathsf{n}))$.

The above restrictions on the way τ acts on encryptions and other primitives (the fifth bullet of the definition) reflect two assumptions on message contents: namely, that they are *deterministic* and *unique*. The "if" part of the statement reflects determinism, meaning that 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 assuming deterministic messages, we can distinguish the case where an actor observes two different randomised encryptions with the same input from the case where he observes the same randomised encryption twice; in the latter case, we will allow an actor to draw certain conclusions from this. The "only if" part reflects uniqueness. Concerning uniqueness, note that differently-constructed messages could a priori 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. We assume that this does not happen, i.e., elements of our grammar at the contents layer uniquely represent message contents (the *structural equivalence* assumption [99]).

The complete knowledge of an actor is modelled by a *knowledge base*. We model this knowledge at the context layer so that we can later determine what knowledge of personal information follows from it. Formally:

Definition 4 Let $I = (\mathcal{L}^c, \mathcal{L}, \mathsf{E}^c, \mathcal{E}, \Leftrightarrow, \sigma, \tau, \{\psi_1, \ldots, \psi_k\})$ be an information model. A *knowledge base* on I is a set $\mathcal{C} \subset \mathcal{L}^c \cup \mathsf{E}^c$.

In addition to the messages an actor has sent and received, his knowledge base needs to contain the pieces of personal information from his initial view. This includes context entities: because they cannot occur in messages, we mention them explicitly in the definition. Also, the knowledge base should contain other relevant material such as secret keys known by the actor, and nonces he has generated during the execution of the cryptographic protocols. Note that we do not need to specify the order of messages: because we use contexts, we can already distinguish between messages from different protocol instances. We use the notation $\mathcal{C}_a$ to refer to the knowledge base of an actor a, and $\mathcal{C}_A$ to refer to the knowledge base of coalition $A \subset \mathcal{A}$ (defined to be the union of the knowledge bases of the respective actors, see Section 4).

In the next example, we show several context messages and the knowledge base of an actor after communication.

Example 4 We consider the PI model of Example 1. We model two context messages in domain π, which represents a protocol instance. First, we model a symmetric encryption of Alice's identifier, encrypted using a shared key. The shared key is modelled by a non-personal item with context-layer representation $shkey|^{\pi}_{\cdot}$. The encryption is then denoted

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

Second, we model a message representing an encryption under $shkey|^{\pi}_{\cdot}$ of Alice's age and a randomised 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 $n|^{\pi}_{\cdot}$. The secret key of the server is context identifier $k^-|^{\pi}_{srv}$. The second message is:

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

We now consider the knowledge base of the client, supposing that he has observed (i.e., sent or received) messages m_1 and m_2. We model the communication addresses that the client and the server have used as context identifiers $ip|^{\pi}_{cl}$, $ip|^{\pi}_{srv}$. The client knows these, as well as messages $\mathsf{m}_1, \mathsf{m}_2$. In addition, his knowledge base contains the personal and other information known at the beginning of the scenario. Apart from his address book, we assume that this initial knowledge includes the shared key, as well as his own address and the address and public key of the server, known in some arbitrary contexts $*|^{\cdot}_{\cdot}$, $*|^{\cdot}_{me}$, $*|^{\cdot}_{srv}$. His full knowledge base after communication is then:

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

with $ds|^{ab}_{12}, ds|^{ab}_{4}$ context entities and the other elements of $\mathcal{C}_{cli}$ context messages. □

3.2 From Knowledge Base to View

In this subsection, we show how to determine the view corresponding to a knowledge base. The first component of the view, the set of detectable items, is defined using a deductive system. The second component, the associability relation, is defined based on linking identifiers and entities.

3.2.1 A Deductive System for Detectability

In this section, we define what messages containing personal information can be built from knowledge base $\mathcal{C}$. Informally, we say that message m is *detectable* from $\mathcal{C}$, written $\mathcal{C} \vdash \mathsf{m}$, if it can be obtained from messages in $\mathcal{C}$ using the three operations on messages described at the beginning of this section: reading information from them, applying cryptographic operations on them, and comparing the contents of messages. In particular, detectability of context identifiers and data items in a view is defined as detectability from as messages from the corresponding knowledge base.

The semantics of $\vdash$ is given by a deductive system. Deductive systems are commonly used in protocol analysis to reason about what messages an attacker can fabricate (see, e.g., [37,49]). Typically, such deductive systems consist of general *axioms* stating which messages are known; and particular *construction* and *elimination* rules stating the functionality of cryptographic primitives: construction rules describe how a cryptographic primitive is constructed from its parts (e.g., a symmetric encryption is constructed from the key and plaintext); *elimination* rules describing how parts can be obtained from a cryptographic primitive by applying cryptographic operations (e.g. the plaintext is obtained from an encryption by decrypting using the key). However, in these works, such rules operate directly on message contents, without taking into account what information they represent, or in which context this information is known (i.e., they operate at our contents layer). Conversely, for our purposes, we need to consider the context: hence we need to reinterpret these rules at the context layer and add additional ones. Our formal definition of $\vdash$ is as follows:

Definition 5 Let $\mathcal{C}$ be a knowledge base, and m a context message. The *detectability* relation $\mathcal{C} \vdash \mathsf{m}$ is defined by the inference rules given in Figure 4.

The deductive system in Figure 4 consists of three general rules **($\vdash$0)**, **($\vdash$Eψ)**, and **($\vdash$C)**; and particular *construction*, *elimination*, and *testing* rules for the particular cryptographic primitives modelled. Hence, when using our framework to analyse a system, **($\vdash$0)**, **($\vdash$Eψ)**, and **($\vdash$C)** are always the same; the other rules need to be adapted to model the particular primitives used in the system.

($\vdash$0) is the standard axiom to detect known messages. Construction and elimination rules, in particular those for (standard) hashes, (a)symmetric encryption, concatenation, and signatures, are as usual [37]. For instance, rule **($\vdash$CE)** states that symmetric encryption $E'_{\mathsf{n}}(\mathsf{m})$ can be detected if m and n can be detected, and rule **($\vdash$EE)** states that plaintext m can be obtained from encryption $E'_{\mathsf{n}}(\mathsf{m})$ if key n is known. However, note that because our deductive system operates at the context layer, rule **($\vdash$EE)** only applies if the key is known *in the same context as the message*. Thus, this rule fails to capture that an actor can perform decryption using keys he knows from different contexts. To avoid this problem, we introduce testing rules. These rules let an actor detect a new context-layer representation of a messages whose contents he already knew by applying a cryptographic operation. For instance, rule **($\vdash$TE)** states that if an actor can detect encryption $E'_{\mathsf{n}}(\mathsf{m})$ and any message content equivalent to n, then he can detect n. He can then use n to decrypt the message.

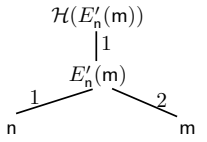


Fig. 6 Parse tree of message $\mathcal{H}(E'_{\mathsf{n}}(\mathsf{m}))$ (left) and submessages (right)

Example 5 shows a typical example of the use of testing and elimination rules.

Example 5 Consider knowledge base $\mathcal{C}_{cli}$ from Example 4. Then $id|^{\pi}_{su}$ is detectable from $\mathcal{C}_{cli}$ by the derivation shown in Figure 5. The derivation models the actor testing whether $skey|_{:}$ is the decryption key for $E'_{shkey|_{:}}(id|_{su})|^{\pi}$ **($\vdash$TE)**. (Rule **($\vdash$TE)** can be applied because $shkey|^{\pi}_{:} \doteq skey|_{:}$.) After learning that it is, the actor can decrypt the message **($\vdash$EE)**. □

We assume that for any cryptographic operation modelled by an elimination rule, there is a corresponding testing rule. (This is an over-estimation in case the actor cannot distinguish between a failed and successful cryptographic operation, e.g. when certain kinds of encryption schemes are used in which the plaintext resulting from decryption cannot be recognised as valid.) On the other hand, not all testing rules testing have a corresponding elimination rule, e.g., rule **($\vdash$TS)** for signature verification.

Differently from other deductive systems, we introduce two additional general rules: **($\vdash$Eψ)** to reason about properties of attributes and **($\vdash$C)** to reason about contents of messages. Rule **($\vdash$Eψ)** states that any properties that apply to an attribute can be detected from the attribute. Note that, because the rule only applies if image $\psi_i(\mathsf{d})$ under the partial function ψ_i is defined, only applicable properties can be detected. Rule **($\vdash$C)** covers knowledge obtained by comparing contents of different messages.

Before we can discuss rule **($\vdash$C)** in detail, we first need to formalise the notion of *submessages* of a message. A message m has a natural syntactic structure according to the grammar in Figure 4. This structure can be represented by a parse tree, in which the nodes are the submessages of m; the root is the message m itself. We write $\mathsf{m}@z$ for the submessage of m at path z from m in its parse tree; the empty path is denoted ϵ. Figure 6 shows the parse tree of a message (left) and the corresponding formal representation of its submessages (right).

If two context messages m_1 and m_2 are content equivalent, then by the uniqueness assumption, their respective

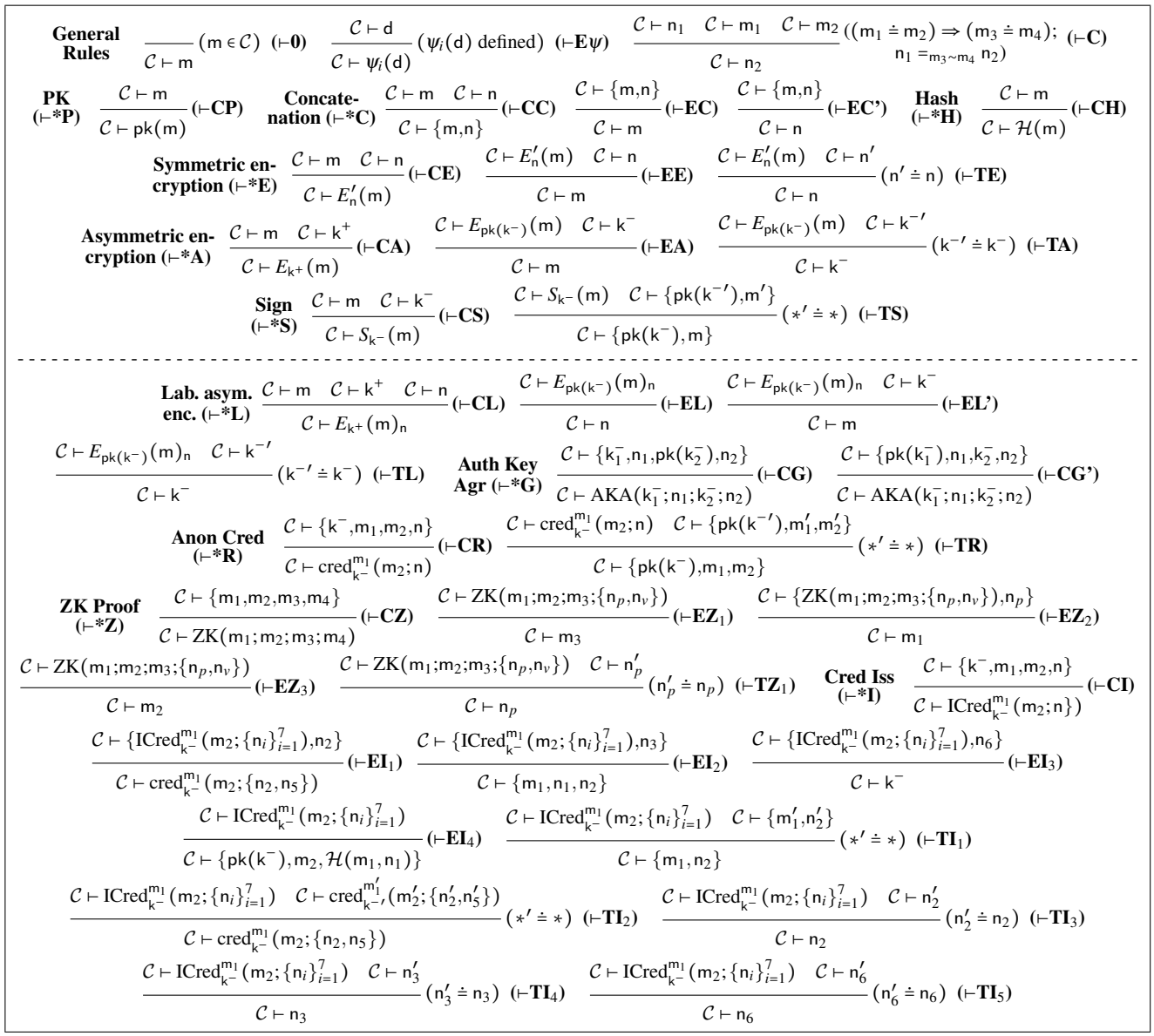


Fig. 4 Deductive system for detectability: m, m_i, n, n_i, k^-, k^+, k_i^- and $*'$ any context message; $\mathsf{d} \in \mathsf{I}^c \cup \mathsf{D}^c$ any context identifier or data item; $\mathsf{p}_* \in \mathsf{D}^c$ any context data item. $*' \doteq *$ means "for any pair of dashed and non-dashed context messages". Rules (⊢**0**) to (⊢**TS**) explained in Section 3.2.1; rules (⊢**CL**) to (⊢$\mathbf{TI}_6$) in Section 3.2.2.

$$\dfrac{\dfrac{}{\mathcal{C}_{cli} \vdash E'_{shkey|_\cdot}(id|_{su})|^{\pi}}(\vdash\mathbf{0}) \quad \dfrac{\dfrac{}{\mathcal{C}_{cli} \vdash E'_{shkey|_\cdot}(id|_{su})|^{\pi}}(\vdash\mathbf{0}) \quad \dfrac{}{\mathcal{C}_{cli} \vdash shkey|_\cdot^{\pi}}(\vdash\mathbf{0})}{\mathcal{C}_{cli} \vdash shkey|_\cdot^{\pi}}(\vdash\mathbf{TE})}{\mathcal{C}_{cli} \vdash id|_{su}^{\pi}}(\vdash\mathbf{EE})$$

Fig. 5 Derivation of $id|_{su}^{\pi}$ given knowledge base $\mathcal{C}_{cli}$ from Example 4 (see Example 5)

submessages are also content equivalent. That is, if $\mathsf{m}_1 \doteq \mathsf{m}_2$ and $\mathsf{m}_1@z$ and $\mathsf{m}_2@z$ are defined (i.e., there exists a submessage at path z), then $\mathsf{m}_1@z \doteq \mathsf{m}_2@z$. Also, if m_1 and m_2 contain data items satisfying a property ψ_k, the content equivalence of that property is also implied. Formally:

Definition 6 The pair $(\mathsf{m}_1, \mathsf{m}_2)$ is *evidence* for $\mathsf{n}_1 \doteq \mathsf{n}_2$, denoted $(\mathsf{m}_1 \doteq \mathsf{m}_2) \Rightarrow (\mathsf{n}_1 \doteq \mathsf{n}_2)$, if one of the following two conditions holds:

- $\mathsf{m}_1 \doteq \mathsf{m}_2$, and for some z, $\mathsf{m}_1@z = \mathsf{n}_1$ and $\mathsf{m}_2@z = \mathsf{n}_2$;
- $(\mathsf{m}_1 \doteq \mathsf{m}_2) \Rightarrow (\mathsf{n}'_1 \doteq \mathsf{n}'_2)$, and for some i, $\mathsf{n}_1 = \psi_i(\mathsf{n}'_1)$ and $\mathsf{n}_2 = \psi_i(\mathsf{n}'_2)$.

The "content analysis" inference rule (⊢**C**) then states that if an actor can derive evidence $(\mathsf{m}_1, \mathsf{m}_2)$ for $\mathsf{n}_1 \doteq \mathsf{n}_2$ and he can derive a message with n_1 in it, then he can derive the same message with n_1 replaced by n_2, and vice versa. The following example shows how (⊢**C**) models an actor determining a piece of information by reasoning about its contents:

Example 6 Consider knowledge base

$$\mathcal{C}_a = \{\mathcal{H}(\{id, age\})|_1^\eta, id|_2^\eta, age|_3^\eta\},$$

where $id|_1^\eta \doteq id|_2^\eta$ and $age|_1^\eta \doteq age|_3^\eta$. Intuitively, an actor can learn $id|_1^\eta$ from this knowledge base: he can construct the hash $\mathcal{H}(\{id|_2^\eta, age|_3^\eta\})$, note that it has the same contents as $\mathcal{H}(\{id, age\})|_1^\eta$, and thus infer that $id|_1^\eta$ must have the same contents as $id|_2^\eta$, which he knows.

This series of reasoning step is captured in the derivation shown in Figure 7. Namely, $\mathcal{C}_a \vdash \mathcal{H}(\{id, age\})|_1^\eta$ holds, and by (⊢**CC**), (⊢**CH**) we have $\mathcal{C}_a \vdash \mathcal{H}(\{id|_2^\eta, age|_3^\eta\})$. By Definition 6, the pair $(\mathcal{H}(\{id, age\})|_1^\eta, \mathcal{H}(\{id|_2^\eta, age|_3^\eta\}))$ is evidence for $id|_1^\eta \doteq id|_2^\eta$ (as well as for $age|_1^\eta \doteq age|_3^\eta$). By (⊢**C**), he can then deduce $id|_1^\eta$. (In the same way, also $\mathcal{C}_a \vdash age|_1^\eta$ follows.) □

3.2.2 Inference Rules for Non-Standard Primitives

We now discuss the inference rules for the non-standard primitives modelled in this paper. Labelled asymmetric encryption is similar to normal asymmetric encryption; note that the label can be derived from the encryption (⊢**EL'**), but to change it, the plaintext is needed, i.e., the label is unmodifiably attached. To derive a session key using authenticated key agreement, an actor needs to know one of the private keys used, the other public key, and both parties' randomness (⊢**CG**), (⊢**CG'**).

Messages $\mathrm{ZK}(\ldots)$ and $\mathrm{ICred}_\star^*(\ldots)$ represent the complete transcripts of instances of zero-knowledge proofs and credential issuing protocols, respectively. In particular, the construction rules for these messages state which inputs are required to build the complete transcript. (When such a protocol is run, two different parties each provide part of the input; this is captured by traces, see Section 4.)

The inference rules for ZK given in our deductive system model the privacy aspects of a large family of ZK proofs known as "Σ-protocols" [41]. Σ-protocols exist for many properties; in particular, they are used to prove properties of anonymous credentials [8]. The randomness for Σ-protocols is of the form $\{\mathsf{n}_p, \mathsf{n}_v\}$, representing contributions by the prover and verifier, respectively. Apart from the usual construction rule, there are three elimination rules: (⊢$\mathbf{EZ}_1$) states that the property proven by a ZK proof can be seen from its transcript; (⊢$\mathbf{EZ}_2$) states that the prover's secret can be derived from the prover's randomness; and (⊢$\mathbf{EZ}_3$) states that the public information can be derived. Testing rule (⊢$\mathbf{TZ}_1$) can be applied to obtain the prover's randomness. We assume that parties do not reuse their randomness; also, because we are only interested in privacy aspects, we only consider deriving randomness if that randomness can be used to derive other information. See Appendix B.1 for details.

The inference rules for cred and ICred model anonymous credentials and their issuing protocol based on SRSA-CL signatures [8]. Anonymous credentials can be verified to correspond to a given verification key, message and secret identifier (⊢**TR**). Anonymous credentials are usually derived from the transcript of its issuing protocol (⊢$\mathbf{EI}_1$) by the user (the issuer does not know n_2 and so does not learn the credential); but they can also be constructed directly from its components (⊢**CR**). Similarly for the issuing protocol transcript itself (⊢**CI**). Before the issuing protocol takes place, the user needs to have sent a randomised commitment $\mathcal{H}(\mathsf{m}_1, \mathsf{n}_1)$ to her secret identifier to the issuer. During the protocol, additional randomness $\mathsf{n}_2, \ldots, \mathsf{n}_8$ is generated by the two parties; $\mathsf{n}_1, \ldots, \mathsf{n}_8$ together form the randomness component of the ICred primitive. Inference rules (⊢$\mathbf{EI}_2$) and (⊢$\mathbf{EI}_3$) model the inference of secret information from the transcript using randomness. A credential issuing protocol transcript allows for deriving and testing of various nonces and information used ((⊢$\mathbf{EI}_4$); (⊢$\mathbf{TI}_1$)–(⊢$\mathbf{TI}_5$)). As with our model of ZK proofs, we only consider rules needed to infer personal information, and assume non-reuse of randomness. In Appendix B.2 we explain why these rules accurately capture privacy aspects.

3.2.3 Associability and View

Having discussed detectability, we consider the other part of an actor view: associability. We determine the associability relation corresponding to a knowledge base $\mathcal{C}$ by finding out which identifiers and entities are known to be equivalent in $\mathcal{C}$:

Definition 7 Let $\mathcal{C}$ be a knowledge base. The *associability relation* $\leftrightarrow$ *corresponding to* $\mathcal{C}$ is the equivalence relation on O^c obtained by evaluating the following rules:

1. For all $ds|_k^\eta, ds|_l^\zeta \in \mathcal{C} \cap \mathsf{E}^c$: if $\sigma(ds|_k^\eta) = \sigma(ds|_l^\zeta)$, then $ds|_k^\eta \leftrightarrow ds|_l^\zeta$;
2. For all $x|_k^\eta, y|_k^\eta \in \mathsf{O}^c$: $x|_k^\eta \leftrightarrow y|_k^\eta$;
3. If $\mathcal{C} \vdash \mathsf{m}_1, \mathcal{C} \vdash \mathsf{m}_2$, and $(\mathsf{m}_1 \doteq \mathsf{m}_2) \Rightarrow (\mathsf{i}_1 \doteq \mathsf{i}_2)$ for $\mathsf{i}_1, \mathsf{i}_2 \in \mathsf{I}^c$, then $\mathsf{i}_1 \leftrightarrow \mathsf{i}_2$.

and taking the reflexive, symmetric, transitive closure.

The first point states that any known context entities representing the same entity can be associated; the second point states that all information from the same context can be associated. The third point captures associations by identifiers. Actors do not need to be able to detect the identifier: instead, it is sufficient to detect evidence for content equiv-

$$\dfrac{\dfrac{}{\mathcal{C}_a \vdash id|_2^\eta}(\vdash\mathbf{0}) \quad \dfrac{\dfrac{}{\mathcal{C}_a \vdash age|_3^\eta}(\vdash\mathbf{0})}{\mathcal{C}_a \vdash \{id|_2^\eta, age|_3^\eta\}}(\vdash\mathbf{CC})}{\mathcal{C}_a \vdash \mathcal{H}(\{id|_2^\eta, age|_3^\eta\})}(\vdash\mathbf{CH}) \quad \dfrac{}{\mathcal{C}_a \vdash \mathcal{H}(\{id,age\})|_1^\eta}(\vdash\mathbf{0})}{\mathcal{C}_a \vdash id|_1^\eta}(\vdash\mathbf{C})$$

Fig. 7 Derivation of $id|_1^\eta$ given knowledge base $\mathcal{C}_a = \{\mathcal{H}(\{id,age\})|_1^\eta, id|_2^\eta, age|_3^\eta\}$ (see Example 6).

alence (Definition 6). The following example demonstrates the definition.

Example 7 We determine the associability relation $\leftrightarrow_{cli}$ corresponding to the knowledge base $\mathcal{C}_{cli}$ from Example 4:

$$\begin{aligned}\mathcal{C}_{cli} = \{&ds|_{12}^{ab}, teln|_{12}^{ab}, ds|_4^{ab}, id|_4^{ab}, skey|_{\cdot}^{\cdot}, ip|_{me}^{\cdot}, ip|_{srv}^{\cdot},\\ &\mathsf{pk}(k^-|_{srv}^{\cdot}), ip|_{cl}^{\pi}, ip|_{srv}^{\pi}, E'_{shkey|_{\cdot}}(id|_{su})|^\pi,\\ &E'_{shkey|_{\cdot}}(\{age|_{su}, n|_{\cdot}, S_{k^-|_{srv}}(\{age|_{su}, n|_{\cdot}\})\})|^\pi\}.\end{aligned}$$

Rule 2 from Definition 7 allows association of information from the same context; thus, e.g., $ds|_4^{ab} \leftrightarrow_{cli} id|_4^{ab}$. Rule 3 allows association of context identifiers using evidence of content equivalence. For instance, clearly, $\mathcal{C}_{cli} \vdash id|_4^{ab}$, $\mathcal{C}_{cli} \vdash id|_{su}^{\pi}$ (see Example 5), and $(id|_4^{ab} \doteq id|_{su}^{\pi}) \Rightarrow (id|_4^{ab} \doteq id|_{su}^{\pi})$, hence $id|_4^{ab} \leftrightarrow id|_{su}^{\pi}$. In fact, all context items about Alice that occur in $\mathcal{C}_{cli}$ turn out to be mutually associable. However, rule 1 for associating entities does not apply since, e.g., $\sigma(ds|_{12}^{ab}) \neq \sigma(ds|_4^{ab})$. Continuing in this way, the items detectable from $\mathcal{C}_{cli}$ form the following equivalence classes under $\leftrightarrow_{cli}$:

$$\{ds|_{12}^{ab}, teln|_{12}^{ab}\}\ \ \{ds|_4^{ab}, id|_4^{ab}, id|_{su}^{\pi}, age|_{su}^{\pi}\}$$

$$\{ip|_{cl}^{\pi}, ip|_{me}^{\cdot},\}\ \ \{ip|_{srv}^{\cdot}, k^-|_{srv}^{\cdot}, ip|_{srv}^{\pi}, k^-|_{srv}^{\pi}\},$$

with data subjects Bob, Alice, the client, and the server, respectively. □

Note that $\leftrightarrow$ is intentionally defined on all context-layer items, and not just on detectable context-layer items. The following example shows how this broader definition allows additional inferences to be made:

Example 8 Let $\mathcal{C}_a = \{\{E_{shakey|_{\cdot}}(id|_1), d|_1\}|^\eta, \{E_{shakey|_{\cdot}}(id|_1), d'|_1\}|^\chi\}$ be a knowledge base, where $shakey|_{\cdot}^{\eta} \doteq shakey|_{\cdot}^{\chi}$ and $id|_1^\eta \doteq id|_1^\chi$. Let $\leftrightarrow_a$ be the associability relation corresponding to $\mathcal{C}_a$. Intuitively, even if the key used in the encryptions in $\mathcal{C}_a$ is unknown, the fact that they have the same contents means that the two identifiers, and hence also the two data items $d|_1^\eta$, $d'|_1^\chi$, must have the same data subject. Indeed, because

$$(E_{shakey|_{\cdot}}(id|_1)|^\eta \doteq E_{shakey|_{\cdot}}(id|_1)|^\chi) \Rightarrow (id|_1^\eta \doteq id|_1^\chi),$$

rule 3 from Definition 7 gives $id|_1^\eta \leftrightarrow_a id|_1^\chi$; by rule 2 and transitivity, $d|_1^\eta \leftrightarrow_a d'|_1^\chi$. □

We can now define the view corresponding to a knowledge base:

Definition 8 Let $\mathcal{C}$ a knowledge base. The *view V corresponding to $\mathcal{C}$* is the view $V = (\mathsf{O}, \leftrightarrow)$, where $\mathsf{O} = \{\mathsf{p} \in \mathsf{I}^c \cup \mathsf{D}^c \mid \mathcal{C} \vdash \mathsf{p}\} \cup (\mathcal{C} \cap \mathsf{E}^c)$, and $\leftrightarrow$ is as in Definition 7.

Example 9 We determine the view $V_{cli} = (\mathsf{O}_{cli}, \leftrightarrow_{cli})$ corresponding to the knowledge base $\mathcal{C}_{cli}$ from Example 4. First, let us consider the view of the client on Alice and Bob. On Alice, we have $id|_{su}^{\pi} \in \mathsf{O}_{cli}$ because $\mathcal{C}_{cli} \vdash id|_{su}^{\pi}$, as shown in Example 5. Similarly, $ds|_4^{ab}, id|_4^{ab}, age|_{su}^{\pi} \in \mathsf{O}_{cli}$, and as we saw in Example 7, they are mutually associable. On Bob, the two items $ds|_{12}^{ab}$ and $teln|_{12}^{ab}$ are detectable and associable. In fact, the client's view on Alice and Bob is as in Figure 3.

Apart from this, the client's view also contains knowledge about the client and server. Namely, in both cases, it contains two associable context-layer representations of the communication address: $ip|_{me}^{\cdot}, ip|_{cl}^{\pi} \in \mathsf{O}_{cli}$ on the client, and $ip|_{srv}^{\cdot}, ip|_{srv}^{\pi} \in \mathsf{O}_{cli}$ on the server. □

3.3 Deciding Detectability and Linkability

In this section, we present the algorithms to decide detectability and linkability used in our tool. Our tool consists of a series of Prolog scripts[2] for the automatic verification of privacy requirements for a set of architectures. The most technically challenging part of this task is to compute the views of actors (i.e., the sets of detectable items and associability relations) from their knowledge bases. Here, we describe our algorithms and their efficiency in general terms; for details, refer to the documentation of the implementation.

Our deductive system is essentially a traditional deductive system [37,49] to which testing rules and the content analysis rule have been added. Let us first ignore content analysis, and only consider the construction, testing and elimination rules. Construction rules generally derive messages from submessages; testing and elimination rules derive submessages from messages using some "additional prerequisites" (e.g., the key for the decryption rule (⊢**EE**)). As testing/elimination and construction cancel each other out, there is no point in applying testing/elimination to the result of construction rule. Thus, to check the derivability of a message m, we try to find a message n in which it occurs as

[2] The implementation, along with its documentation, can be downloaded at `http://www.mobiman.me/publications/downloads/`.

```
1: {Let ⊨ denote the deductive system without the content analysis rule}
2: for all context items m′: m′ ≐ m, C_a ⊨ m′ do
3:    for all context items p, p′: m@z = p, m′@z = p′, p ≠ p′ do
4:       {Find sequence of evidence for p ≐ p′ using breadth-first search}
5:       Q ← {p} {queue of items to check}; P ← {} {already checked}; found ← false
6:       while Q ≠ {} ∧ ¬found do
7:          q ← pop(Q); P ← P ∪ {q} {move q from queue to already checked}
8:          if q = p′ then found ← true; break {evidence for p ≐ p′ found} end if
9:          for all context items q′: q′ occurs in message in C_a, q′ ≐ q, q′ ∉ P ∪ Q do
10:            {Try to find evidence for q ≐ q′}
11:            for all context items n: C_a ⊨ n, n is minimal w.r.t. q do
12:               if ∃n′ : C_a ⊨ n′ : (n ≐ n′) ⇒ (q ≐ q′) then Q ← Q ∪ {q′} end if
13:            end for
14:            for all context items n′: C_a ⊨ n′, n′ is minimal w.r.t. q′ do
15:               if ∃n : C_a ⊨ n : (n ≐ n′) ⇒ (q ≐ q′) then Q ← Q ∪ {q′} end if
16:            end for
17:         end for
18:      end while
19:      if ¬found then break {No such p′ found: try next m′} end if
20:   end for
21:   return true{Actor has evidence that m ≐ m′ for a m′ such that C_a ⊨ m′}
22: end for
23: return false{For all m′ such that m ≐ m′, C_a ⊨ m′: actor has no evidence for m ≐ m′}
```

Fig. 8 Algorithm implementing the deductive system: given knowledge base $\mathcal{C}_a$ and context message m, check whether $\mathcal{C}_a \vdash \mathsf{m}$

submessage, and try to derive m from it using elimination and testing. If this does not work, we repeat the procedure for m's submessages: if successful, then m can be obtained from them with a construction rule.

While trying elimination or testing rules, we need to check the derivability of the additional prerequisites n. We claim that this check can be done at the contents layer (so a simple deductive system suffices). For the testing rule this is clear; however, it also holds for elimination rules because their additional prerequisites can always be obtained from a content equivalent message using the testing rule.

Thus, in terms of evaluation, our deductive system differs from standard systems in two ways. First, for elimination rules, the additional prerequisites are evaluated not using the deductive system itself, but using a (standard) deductive system at the contents layer. Second, testing rules are added which are evaluated in the same way as elimination rules. Intuitively, our deductive system is thus not much harder to evaluate than a corresponding standard deductive system. (However, typically it will be run on a larger message set because information has multiple representations.)

We now turn our implementation of the deductive system without content analysis into an implementation of the full deductive system. Note that any deduction in the full deductive system can be transformed into a deduction deriving the same message satisfying the following conditions:

- After content analysis rules, no other rules are applied to a message
- In any application of ($\vdash$**C**), the message n_2 and the message n_1 from which it is derived only differ by one context item at one position
- In any application of ($\vdash$**C**), the messages m_1 and m_2 are derived without content analysis; also, m_1 is minimal with respect to n_1 in the sense that no elimination or testing rule can be applied to it to obtain a submessage containing n_1; and/or n_2 is minimal with respect to m_2.

The algorithm in Figure 8 is an imperative translation of our Prolog implementation; by the above properties, it implements derivability in our full deductive system. Namely, to derive m from a given knowledge base $\mathcal{C}_a$, it takes all messages $\mathsf{m}' \doteq \mathsf{m}$ such that $\mathcal{C}_a \vdash \mathsf{m}'$, and tries to obtain m' from m by content analysis in a context-item-by-context-item fashion. For all positions z at which m and m' differ, the algorithm performs a breadth-first search for messages obtained from m by content analysis at position z, until it finds m with $\mathsf{m}@z$ replaced by $\mathsf{m}'@z$. The breadth-first search is performed by first searching for a minimal message using testing and elimination rules (lines 10 and 13); and then searching for a content equivalent message using testing, elimination and construction rules (lines 11 and 14). We did not optimise this algorithm in terms of complexity. Indeed, in practice, most context items are content equivalent only to few other items, so the search space for the algorithm is very limited.

The algorithm for checking the associability of two contexts is similar to the previous algorithm. In particular, it starts with one context (η, k) and uses breadth-first search to find associable contexts. This involves finding all identifiers and entities that occur in (η, k) and all other contexts in which that identifier/entity occurs. The algorithm then searches evidence for content equivalence of the different representations of the identifier/entity.

4 States, Traces, and System Evolution

In this section, we complete our formal framework for the analysis of data minimisation by modelling communication in an information system. In Section 3, we showed how to determine what knowledge of personal information actors have given their knowledge bases. In this section, we show how these knowledge bases, collected in a *state*, can be derived from a model of exchanged messages given as *traces*. (The approach of our framework is to model messages based on protocol descriptions. In an alternative type of analysis, the knowledge base of an actor could be derived from communication logs and then analysed using the methods presented in Section 3.)

A state collects the knowledge of all actors in an information system at a certain point in time. 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 9 Let $\mathcal{A}$ be a set of actors, and I an information model.

- A *state of I involving $\mathcal{A}$* 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 8).
- 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 series of communication steps. Each communication step is modelled by a *message transmission* involving two parties that both use a particular communication address modelled as a context identifier. We consider three types of message transmissions. The simplest type (1) captures an actor using address a to send a message m to another actor using address b. Two other types model the execution of cryptographic protocols: type (2) denotes a zero-knowledge proof with prover using address a and verifier using address b; type (3) denotes a credential issuing protocol with user a and issuer b.

Definition 10 A *message transmission* is of one of the following three types:

$$(1)\ \mathsf{a} \to \mathsf{b} : \mathsf{m};\quad (2)\ \mathsf{a} \mapsto \mathsf{b} : \mathrm{ZK}(\mathsf{m}_1;\mathsf{m}_2;\mathsf{m}_3;\mathsf{m}_4);$$

$$(3)\ \mathsf{a} \mapsto \mathsf{b} : \mathrm{ICred}^{\mathsf{m}_1}_{\mathsf{m}_2}(\mathsf{m}_3;\mathsf{m}_4),$$

with a,b context identifiers, and m_i context messages.

Definition 11 A *trace* $\mathfrak{T}$ is a sequence $\mathsf{t}_1;\cdots;\mathsf{t}_k$ of message transmissions.

States *evolve* by traces so that the actors involved learn the messages exchanged:

Definition 12 An *evolution* from state $\{\mathcal{C}^0_x\}_{x\in\mathcal{A}}$ into state $\{\mathcal{C}^k_x\}_{x\in\mathcal{A}}$ by trace $\mathsf{t}_1;\cdots;\mathsf{t}_k$ is a series of steps (let $\mathsf{t}_i = \mathsf{a}_i \to \mathsf{b}_i : \mathsf{m}_i$ or $\mathsf{t}_i = \mathsf{a}_i \mapsto \mathsf{b}_i : \mathsf{m}_i$):

$$\{\mathcal{C}^0_x\}_{x\in\mathcal{A}} \xrightarrow{\mathsf{t}_1} \{\mathcal{C}^1_x\}_{x\in\mathcal{A}} \xrightarrow{\mathsf{t}_2} \cdots \xrightarrow{\mathsf{t}_n} \{\mathcal{C}^n_x\}_{x\in\mathcal{A}},$$

where for every actor $z \in \mathcal{A}$, $\mathcal{C}^i_z = \mathcal{C}^{i-1}_z \cup \{\mathsf{a}_i, \mathsf{b}_i, \mathsf{m}_i\}$ if $z \leftrightarrow \sigma(\mathsf{a}_i)$ or $z \leftrightarrow \sigma(\mathsf{b}_i)$, and $\mathcal{C}^i_z = \mathcal{C}^{i-1}_z$ otherwise.

The following example demonstrates traces, states, and message transmissions.

Example 10 Consider again the PI model from Example 1, extended into an information model in Example 4. 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 $\{\mathcal{C}^0_x\}_{x\in\mathcal{A}}$ consists of initial knowledge of the client and server. As discussed before, this needs to include all used communication addresses and keys. As in Example 4, for the client we take:

$$\mathcal{C}^0_{cli} = \{ds|^{ab}_{12}, teln|^{ab}_{12}, ds|^{ab}_{4}, id|^{ab}_{4}, skey|^{\cdot}_{\cdot}, ip|^{\cdot}_{me}, ip|^{\cdot}_{srv}, \mathsf{pk}(k^{-}|^{\cdot}_{srv})\}.$$

Similarly, for the server we define:

$$\mathcal{C}^0_{srv} = \{key|^{db}_{1}, col1|^{db}_{1}, col1|^{db}_{2}, key|^{db}_{2}, n|^{\cdot}_{\cdot}, skey|^{\cdot}_{\cdot}, ip|^{\cdot}_{srv}, k^{-}|^{\cdot}_{srv}\}.$$

($n|^{\cdot}_{\cdot}$ is the nonce from the server's reply.) The communication described in Example 4 is now formalised by trace $\mathfrak{t}$ consisting of the following message transmissions:

$$ip|^{\cdot}_{cli} \to ip|^{\cdot}_{srv} :\ E'_{shkey|_{\cdot}}(id|_{su})|^{\pi};$$
$$ip|^{\cdot}_{srv} \to ip|^{\cdot}_{cli} :\ E'_{shkey|_{\cdot}}(\{age|_{su}, n|^{\cdot}_{\cdot}, S_{k^{-}|_{srv}}(\{age|_{su}, n|^{\cdot}_{\cdot}\})\})|^{\pi}.$$

Then, state $\{\mathcal{C}^0_x\}_{x\in\mathcal{A}}$ evolves by $\mathfrak{t}$ into state $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$, where:

$$\begin{aligned}\mathcal{C}_{srv} =& \mathcal{C}^0_{srv} \cup \{ip|^{\pi}_{cli}, ip|^{\pi}_{srv}, E'_{shkey|_{\cdot}}(id|_{su})|^{\pi},\\ & E'_{shkey|_{\cdot}}(\{age|_{su}, n|^{\cdot}_{\cdot}, S_{k^{-}|_{srv}}(\{age|_{su}, n|^{\cdot}_{\cdot}\})\})|^{\pi}\},\\ \mathcal{C}_{cli} =& \mathcal{C}^0_{cli} \cup \{ip|^{\pi}_{cli}, ip|^{\pi}_{srv}, E'_{shkey|_{\cdot}}(id|_{su})|^{\pi},\\ & E'_{shkey|_{\cdot}}(\{age|_{su}, n|^{\cdot}_{\cdot}, S_{k^{-}|_{srv}}(\{age|_{su}, n|^{\cdot}_{\cdot}\})\})|^{\pi}\}.\end{aligned}$$

Note that $\mathcal{C}_{cli}$ is as in Example 4. The views of *cli*, *srv* and the coalition $\{cli, srv\}$ about Alice and Bob in this state are as shown in Figure 3. □

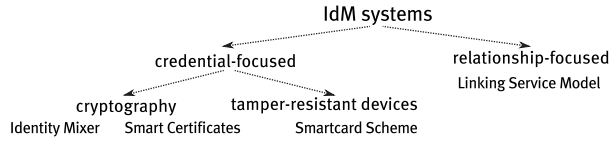


Fig. 9 Taxonomy of IdM systems

5 Case Study: Privacy in Identity Management Systems

Having described our privacy comparison framework, we now introduce a case study to demonstrate its operation. In Sections 2–4, we have presented the various formalisms needed to perform the four steps of our privacy comparison framework (Figure 1). In the case study, we will demonstrate these four steps by comparing the data minimisation characteristics of several identity management (IdM) systems. In this section, we introduce the case study. First, we provide an overview of IdM systems (§5.1). Then, we discuss the requirements related to privacy by data minimisation that are relevant for IdM systems (§5.2); and present the four IdM systems we analyse (§5.3).

5.1 Identity Management Systems

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 management model, called the *isolated user identity management model* [57], 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.

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. (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 scope of this work.) 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.

IdM systems can be divided into two main categories [11] depending on whether or not the identity providers are involved in the service provision phase: *credential-focused* and *relationship-focused* systems (also know as network-based and claim-based systems [3]). Figure 9 shows a taxonomy of IdM systems.

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 can 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 [72], U-Prove [75] and Identity Mixer [8]. The system presented in [100] relies on the use of a smartcard as a tamper-resistant device.

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* [31]) 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 management* [57,87]. (Note that this term is also used to describe the general concept of sharing information between different domains [3] or the mere use of multiple identity providers [1]. To avoid confusion, we will not use it further.) Relationship-focused systems include Liberty Alliance [54], Shibboleth [47], and the linking service model [31].

Because in IdM systems, large amounts of personal information are processed by many different parties, privacy has become a major concern [53,90]. In such systems, privacy threats posed by authorised insiders are nowadays considered to be a critical problem besides outsider attacks on cryptographic protocols [52]. 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* [90]. 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 [90] to compile even more comprehensive profiles. *Privacy-enhancing IdM systems* (e.g., [8,31,100]) aim to minimise the amount of information disclosed as well as prevent that different pieces of information can be linked together [53].

5.2 Requirements

We now present a set of privacy requirements for IdM systems. We have elicited these requirements by analysing the information that actors can learn; considering which of this knowledge should be avoided; and systematically grouping this knowledge into requirements according to what kind of knowledge it is, and who should or should not learn it. We validate our set of requirements in two different ways. First, we check if they cover relevant privacy requirements discussed in the literature. For this, we have studied taxonomies of privacy in identity management [11,53] and the proposals for the identity management systems analysed in this paper [8,31,100], and verified if all requirements discussed in these works are covered by our requirements. 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 requirements expressible in our model, checked which of these can lead to privacy risks, and verified that the relevant ones are covered by our requirements.

Table 2 lists our privacy requirements, also showing in which existing works they are discussed. We first present our requirements, then discuss if they cover all relevant requirements from the literature mentioned above. The analysis of coverage of situations expressible in our model is presented in Section 6.2.

The basic *functional requirement* for IdM systems is that the service provider learns the attributes it needs [12]: *attribute exchange* (AX). Note that in one service provision, a service provider may need attributes from several identity providers.

Privacy requirements cover that certain personal information should not be learned by certain actors. Privacy by data minimisation attempts to minimise the amount of information learned, and the extent to which it can be linked together [53]. The first aspect, information learned, can be further divided into explicitly and implicitly transferred information [90]. *Detectability* requirements capture explicitly transferred information: information about the user's attributes. *Involvement* requirements capture information about whether actors know about each other's involvement with the user: a kind of implicitly transferred personal information. The second aspect is captured in *linkability* requirements: namely, requirements that (combinations of) parties should be able to link personal information from different sessions, databases, etc. as little as possible.

We define three detectability requirements. The first are about the service provider learning no more than strictly necessary: no attribute that he does not need to know (*irrelevant attribute undetectability*, SID), and no complete attribute value if all he needs to know is whether or not an attribute satisfies a certain property [8] (*property-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 property of attributes stored at other identity identity providers: we call this requirement *IdP attribute undetectability* (ID).

Involvement requirements 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 [86] 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. 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, one can aim to achieve that identity providers do not know the identity of other identity providers the user has an account at [31]: 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 requirement *IdP-SP involvement undetectability* (ISM).

Linkability is another fundamental privacy concern because it determines what user profiles can be constructed from the data that is collected [79]. 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.

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 [53]. 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.

An *accountability requirement* counterbalances the privacy guaranteed by the ISL requirement. Namely, if the user misbehaves, it should be possible to identify her [8]. Several IdM systems [8,100] introduce a trusted third party that, in such cases, can help with the identification. The *anonymity revocation* (AR) requirement states that, possibly with the help of this trusted third party, the service provider and iden-

Functional requirements	**Description**	**References**
Attribute exchange (AX)	The service provider learns the value of the required attributes/properties of the user requesting the service.	[8,12,31,77,100]
Privacy requirements	**Description**	
Irrelevant attribute undetectability (SID)	The service provider does not learn anything about attribute values irrelevant to the transaction.	[8,12,77,100]
Property-attribute undetectability (SPD)	The service provider does not learn anything about attributes apart from the properties he needs to know.	[8,12,77,100]
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.	[31]
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.	[8,12,53,31,100]
IdP service access unlinkability (IL)	Identity providers cannot link service access to the user profile they manage.	[53]
IdP profile unlinkability (IIL)	Collaborating identity providers cannot link user profiles.	[53,100]
IdP-SP unlinkability (ISL)	Identity providers and service provider cannot link service accesses to user profile at identity provider.	[8,53,100]
Accountability requirements	**Description**	
Anonymity revocation (AR)	Service provider and identity providers (possibly with help from trusted third party) can reconstruct link between service access and user profile.	[8,12,53,100]

Table 2 Requirements for IdM systems

tity 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.)

When comparing our requirements to those found in existing taxonomies [12,53], we find that our requirements are generally more detailed. In [12], three requirement on data minimisation are presented: conditional release, selective disclosure, and unlinkability. These three requirements correspond to anonymity revocation and IdP-SP unlinkability; irrelevant attribute and property-attribute undetectability; and session unlinkability, respectively (for selective disclosure, the authors do not distinguish between attributes and properties). The authors also mention policy support, which we do not cover. On the other hand, our other requirements are not addressed. In [53], "user-controlled linkage of personal data" is mentioned as the underlying principle of privacy-enhancing identity management. This includes our unlinkability properties (although [53] does not identify them separately), but also a "control" aspect of privacy which we do not cover. The authors of [53] also stress that the desired degree of linkability depends on the application, mentioning revocation in particular.

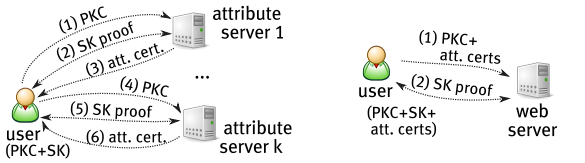
As shown in the table, many of our requirements are discussed by designers of IdM systems [8,31,100]. We compare our requirements to those claimed by designers (including the ones we do not cover) when discussing these systems in Section 5.3.

5.3 Four Systems

We now present the four IdM systems we formally analyse. We consider one traditional system, *smart certificates* [77], 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* [31], a relationship-focused IdM system; *Identity Mixer* [8], a credential-focused system relying on cryptographic protocols; and a credential-focused IdM system based on smartcards [100] we will refer to as the *Smartcard scheme*.

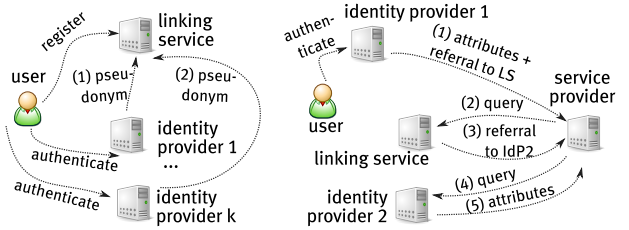
For our analysis, we aimed to cover differ kinds of IdM systems that exist in the literature. In particular, this means selecting credential-focused and relationship-focused systems [3,12]. For the former type, Identity Mixer has received a lot of attention in the research community. For the latter type, the linking service is one of few proposals supporting multiple identity providers that takes privacy into account [31]. We then also included 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 informally discuss to what extent they generalise to similar systems.

We now briefly discuss these systems and the privacy guarantees that they have been designed to provide.



(a) Registration phase (b) Service provision phase

Fig. 10 Smart certificates



(a) Registration phase (b) Service provision phase

Fig. 11 Linking service model

5.3.1 Smart Certificates

Park et al. [77] proposed 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 that link attributes to the public key certificate. In our analysis, we consider one particular variant described in [77]: the user-pull model with long-lived certificates obtained during registration.

The flow of information is summarised in Figure 10. In the registration phase (Figure 10(a)), the user gets an attribute certificate from an identity provider (the "attribute server" in [77]), 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 10(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 [77] is mainly designed to satisfy the attribute exchange requirement (AX) 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.

5.3.2 Linking Service Model

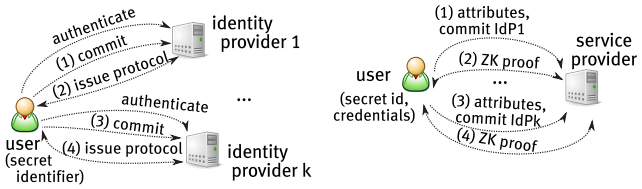
The linking service model [31] is a relationship-focused IdM system. Its main goal is to facilitate the collection of user attributes from different identity providers in a privacy-friendly way without the user having to authenticate to each identity provider separately. To this end, this model includes 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 11. During registration (Figure 11(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 11(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 requirement (AX) as well as a number of privacy requirements [31]. 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 requirement *LS attribute undetectability* (LD); it is not listed in Table 2 because it is only relevant for this system; however, our analysis will include the verification of this requirement.

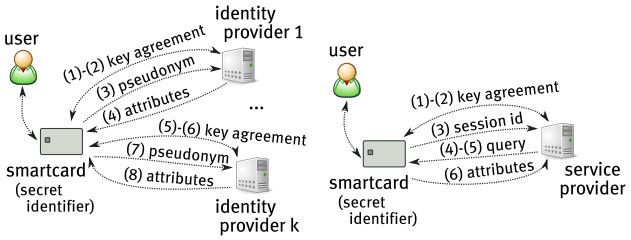
5.3.3 Identity Mixer

Identity Mixer [8] is a credential-focused IdM system using a cryptographic primitive called anonymous credentials. These credentials link attributes to a user identifier, but are



(a) Registration phase (b) Service provision phase

Fig. 12 Identity Mixer



(a) Registration phase (b) Service provision phase

Fig. 13 Smartcard scheme

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 the same user, while integrity of the attributes is still assured.

Figure 12 shows the information flows in Identity Mixer. During registration (Figure 12(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 12(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 requirements [8]. In particular, it aims to satisfy both session unlinkability and IdP/SP unlinkability (together called "multi-show unlinkability" in [8]) and irrelevant attribute and property-attribute undetectability (together called "selective show of data items" in [8]). 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 [8]) that can be used for anonymity revocation. Apart from the data minimisation requirements we defined, the system additionally allows credential issuing where an identity provider copies attributes from another certificate without knowing their values ("blind certification" in [8]). The main motivation for this functionality comes from the use of these certificates for e-cash [8]. In traditional identity management scenarios, such as ours, identity providers should know the attributes they endorse, so we do not consider this requirement in this work.

5.3.4 Smartcard Scheme

Vossaert et al [100] proposed a credential-focused IdM system which relies on PKI for authentication and on smartcards (or other tamper-resistant devices) to ensure that attributes are not modified and 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 defined in the scheme is shown in Figure 13. In the registration phase (Figure 13(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 13(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 he 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 requirements related to the knowledge of personal information [100]. The requirements specified correspond to our notions of attribute exchange, session unlinkability, and anonymity revocation. Irrelevant property and property-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.

5.3.5 Privacy Requirements Claimed by Systems

Table 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 requirements

Scheme	AX	AR	SID	SPD	ID	IM	ISM	SL	IL	IIL	ISL
Smart certificates	✓										
Linking service model	✓					✓		✓			
Identity Mixer	✓	✓	✓	✓				✓			✓
Smartcard scheme	✓	✓	✓	✓				✓		✓	✓

Table 3 Comparison of privacy requirements claimed by the various systems

in order to achieve a comprehensive comparison of their privacy features.

6 Formal Analysis of the Case Study

In this section, we formally analyse and compare the IdM systems presented in the previous section. To make this comparison, we perform the four steps of our privacy analysis framework (Figure 1). In Section 6.1, we model the personal information in a scenario (step 1). In Section 6.2, we model the privacy requirements (step 2; we also discuss whether the requirements identified in Section 5.2 cover all privacy risks expressible in our model). In Section 6.3, we model the communication in each IdM system (step 3). In Section 6.4, we verify which requirements hold in which system, and analyse the results (step 4).

6.1 Step 1: Model Personal Information in Scenario

Step 1 of our analysis method is to model the 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 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 of which only a property 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 the scenario to look as realistic as possible.

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 service provider, for statistical purposes, demands to know the city that Alice comes from. Moreover, the e-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. (Note that the scenario does not cover the separate issue of anonymous payment of the e-book.)

Our formalisation of this scenario as (views on) a PI model is shown schematically in Figure 14. Figure 14(a) lists the actors/entities in the system (in this case study, actors and entities coincide). The trusted third party ttp is included because of the anonymity revocation requirement; however, note that it only occurs in the Identity Mixer and Smartcard schemes.

Figures 14(b) and 14(c) summarise the contexts we use to model different representations of personal information. Figure 14(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. 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. Figure 14(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.

Figures 14(d) and 14(e) define the pieces of personal information in the scenario, and the knowledge about them that actors hold in the initial state. For simplicity, we give an explicit context-layer representation, and use some notational conventions to implicitly describe the information and contents layers. Namely, when context items about the same entity using the same variable are denoted in the nor-

$e \in \mathcal{E}$	Actor/entity
al	Alice
ii	Address provider
is	Subscription provider
bs	E-book store
ttp	Trusted third party

(a) Actors/entities

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

Entity	Domains $\cdot,\iota,\kappa,\mu$	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 ii, is, bs, ttp: anybody knows identifiers and public keys $\mathsf{pk}(k^-|^{\cdot}_{*})$; actor knows own private key

Info	al	ii	is	bs	Description
al	$\{al|^{\iota}_{al}, al|^{\zeta}_{u}, al|^{\xi}_{u}\}$	-	-	-	Entity
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>60	$d_2{>}60|^{\iota}_{al}$	$d_2{>}60|^{\kappa}_{al}$	-	-	age "> 60" property
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
$\mathbf{d_7}$	-	-	-	$\{\mathbf{d_7}|^{\zeta}_{\mathbf{u}}, \mathbf{d_7}|^{\xi}_{\mathbf{u}}\}$	Transaction details
$\mathbf{ip}$	$\{\mathbf{ip}|^{\pi}_{\mathbf{u}}, \mathbf{ip}|^{\eta}_{\mathbf{u}}, \mathbf{ip}|^{\zeta}_{\mathbf{u}}, \mathbf{ip}|^{\xi}_{\mathbf{u}}\}$	-	-	-	IP address

(e) Information about Alice: each row is a piece of information (for $\mathbf{d_7}$ and $\mathbf{ip}$: different pieces of information in each domain); columns al, ii, is, and bs show the initial knowledge of actors about the information

Fig. 14 Schematic representation of PI model and initial actor views

mal font (e.g. $i_{ii}|^{\pi}_{u}$ and $i_{ii}|^{\kappa}_{al}$), they are equivalent; when denoted in boldface (e.g. $\mathbf{ip}|^{\pi}_{\mathbf{u}}$, $\mathbf{ip}|^{\eta}_{\mathbf{u}}$), they are all pairwise non-equivalent. 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 attribute property ψ_1 representing if an age is over 60. At the information layer, $\psi_1(d_2) = d_2{>}60$; at the context layer, $\psi_1(d_2|^{\pi}_{u}) = d_2{>}60|^{\pi}_{u}$, and similarly for other contexts.

Figure 14(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 14(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 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 already taken place, 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 $\mathbf{d_7}|^{\zeta}_{\mathbf{u}}, \mathbf{d_7}|^{\xi}_{\mathbf{u}}$ 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 the two service provisions, Alice knows that she is the data subject ($al|^{\zeta}_{u}$, $al|^{\xi}_{u}$); the service provider knows transaction details $\mathbf{d_7}|^{\zeta}_{\mathbf{u}}$, $\mathbf{d_7}|^{\xi}_{\mathbf{u}}$. Alice knows her own IP address $\mathbf{ip}|^{*}_{\mathbf{u}}$, where $* \in \{\pi, \eta, \zeta, \xi\}$; note that it is assumed to change dynamically between sessions.

6.2 Step 2: Model Privacy Requirements

Step 2 of our framework is to formalise the requirements from Table 2 in terms of actor views. As above, the view of an actor $a \in \mathcal{A}$ and a coalition $A \subset \mathcal{A}$ are denoted $V_a = (\mathsf{O}_a, \leftrightarrow_a)$ and $V_A = (\mathsf{O}_A, \leftrightarrow_A)$, respectively. The formalisation of our requirements in terms of these views is shown in Table 4. AX and AR are detectability and linkability requirements (see Section 2.3), respectively. (For AX, note that bs can always associate the personal information of the user to the purchase because of the common context (ζ, u) or (ξ, u), so we do not check this.) SID, SPD and SID are undetectability requirements; SL, IL, IIL, and ISL are unlinkability require-

Requirement	Formalisation
Attribute exchange (AX)	$d_1\vert_u^\zeta, d_2{>}60\vert_u^\zeta, d_6\vert_u^\zeta, d_1\vert_u^\xi, d_2{>}60\vert_u^\xi, d_6\vert_u^\xi \in \mathsf{O}_{bs}$
Anonymity revocation (AR)	$*\vert_{al}^\kappa \leftrightarrow_{\{bs,ii,is,ttp\}} *\vert_u^\zeta \leftrightarrow_{\{bs,ii,is,ttp\}} *\vert_u^\xi$
Irrelevant attribute undetectability (SID)	$d_3\vert_*^* \notin \mathsf{O}_{bs} \wedge d_5\vert_*^* \notin \mathsf{O}_{bs}$
Property-attribute undetectability (SPD)	$d_2\vert_*^* \notin \mathsf{O}_{bs}$
IdP attribute undetectability (ID)	$d_1\vert_*^* \notin \mathsf{O}_{is} \wedge d_2\vert_*^* \notin \mathsf{O}_{is} \wedge d_3\vert_*^* \notin \mathsf{O}_{is} \wedge$ $d_2{>}60\vert_*^* \notin \mathsf{O}_{is} \wedge d_5\vert_*^* \notin \mathsf{O}_{ii} \wedge d_6\vert_*^* \notin \mathsf{O}_{ii}$
Mutual IdP involvement undetectability (IM)	$\neg(\exists p : *\vert_{is}^{\cdot} \leftrightarrow_{ii} *\vert_{idp2}^p \wedge *\vert_u^p \leftrightarrow_{ii} *\vert_{al}^\kappa) \wedge$ $\neg(\exists p : *\vert_{ii}^{\cdot} \leftrightarrow_{is} *\vert_{idp1}^p \wedge *\vert_u^p \leftrightarrow_{is} *\vert_{al}^\mu)$
IdP-SP involvement undetectability (ISM)	$\neg(\exists p : *\vert_{bs}^{\cdot} \leftrightarrow_{ii} *\vert_{sp}^p \wedge *\vert_u^p \leftrightarrow_{ii} *\vert_{al}^\kappa) \wedge$ $\neg(\exists p : *\vert_{bs}^{\cdot} \leftrightarrow_{is} *\vert_{sp}^p \wedge *\vert_u^p \leftrightarrow_{is} *\vert_{al}^\mu)$
Session unlinkability (SL)	$*\vert_u^\zeta \nleftrightarrow_{bs} *\vert_u^\xi$
IdP service access undetectability (IL)	$*\vert_{al}^\kappa \nleftrightarrow_{ii} *\vert_u^\zeta) \wedge *\vert_{al}^\kappa \nleftrightarrow_{ii} *\vert_u^\xi \wedge$ $*\vert_{al}^\mu \nleftrightarrow_{is} *\vert_u^\zeta \wedge *\vert_{al}^\mu \nleftrightarrow_{is} *\vert_u^\xi$
IdP profile unlinkability (IIL)	$*\vert_{al}^\kappa \nleftrightarrow_{\{ii,is\}} *\vert_{al}^\mu$
IdP/SP unlinkability (ISL)	$*\vert_{al}^\kappa \nleftrightarrow_{\{ii,is,bs\}} *\vert_u^\zeta \wedge *\vert_{al}^\mu \nleftrightarrow_{\{ii,is,bs\}} *\vert_u^\zeta \wedge$ $*\vert_{al}^\kappa \nleftrightarrow_{\{ii,is,bs\}} *\vert_u^\xi \wedge *\vert_{al}^\mu \nleftrightarrow_{\{ii,is,bs\}} *\vert_u^\xi$

Table 4 Formalisation of requirements in our scenario ($\mathsf{m} \nleftrightarrow_a \mathsf{n}$ means $\neg(\mathsf{m} \leftrightarrow_a \mathsf{n})$; $*$ means for all possible values)

ments. (Un-)detectability requirements are straightforward to formalise; e.g., property-attribute undetectability means undetectability by *bs* of the context item $d_2\vert_p^\delta$ in any context (δ, p). (Un-)linkability requirements translate to contexts not being associable by an actor or coalition. IM and ISM are non-involvement requirements: 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 $idp2$ profile to $\vert_{idp2}^{\cdot}$ and the *u* profile to $\vert_{al}^\kappa$.

We now analyse whether the above privacy requirements cover all privacy risks expressible in our model. To this end, we consider all coalitions and all possible knowledge (in terms of elementary detectability, involvement, and linkability aspects; see Section 2.3); and verify if they represent a privacy risk, and if so, by which privacy requirement they are captured. The result is shown in Table 5. The first group of columns indicates the coalition with respect to which a requirement is defined; the next groups list the detectability, involvement, and linkability aspects that it entails.

First consider detectability requirements. 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. (Undetectability of endorsed attributes would be a requirement for the blind certification [8] feature of the Identity Mixer scheme as discussed in Section 5.3.3.) There are no detectability requirements 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 requirements 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 requirements 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 requirements with respect to the service provider as important. (Among the analysed systems, only the Smartcard scheme would satisfy them.)

Linkability requirements 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 requirement 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, 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 requirement 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.

6.3 Step 3: Model Communication in IdM systems

Step 3 of our framework is to model the communication in the systems we want to analyse (§5.3). For each system, this formalisation consists of two parts. First, we define an initial state $\{\mathcal{C}_a^0\}_{a\in\mathcal{A}}$ capturing the initial knowledge of all actors, extending the knowledge from Figure 14 with respect to the specific system. Second, we define a trace Scenario that models the communication that takes place in the system in the system from the initial state when registration at *ii*, registration at *is*, and two service provisions at *bs*, are consecutively performed.

Requirement	Coalition of...				■: undetectable w.r.t. coalition □: detectable w.r.t. coalition							Involvement unknown			■: unassociable w.r.t. coalition □: associable w.r.t. coalition					
	bs	*ii*	*is*	*ttp*	d_1	d_2	d_2>60	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 5 Schematic overview of the requirements in Table 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

$$\mathcal{C}^0_{al} = (\text{Fig.14}) \cup \{\text{MS}_{k^-|^\cdot_{ca}}(i|^\cdot_{al}, \mathsf{pk}(k^-|^\cdot_u), n_c|^\cdot), k^-|^\cdot_{al}, \mathbf{n}_{\mathbf{z},\mathbf{a}}|^\pi_\cdot, \mathbf{n}_{\mathbf{z},\mathbf{a}}|^\eta_\cdot, \mathbf{n}_{\mathbf{z},\mathbf{a}}|^\zeta_\cdot, \mathbf{n}_{\mathbf{z},\mathbf{a}}|^\xi_\cdot\};$$
$$\mathcal{C}^0_{ii} = (\text{Fig.14}) \cup \{\mathbf{n}_{\mathbf{z},\mathbf{b}}|^\pi_\cdot, n_a|^\pi_\cdot\};$$
$$\mathcal{C}^0_{is} = (\text{Fig.14}) \cup \{\mathbf{n}_{\mathbf{z},\mathbf{b}}|^\eta_\cdot, n_b|^\eta_\cdot\};$$
$$\mathcal{C}^0_{bs} = (\text{Fig.14}) \cup \{\mathbf{n}_{\mathbf{z},\mathbf{b}}|^\zeta_\cdot, \mathbf{n}_{\mathbf{z},\mathbf{b}}|^\xi_\cdot\}$$

$\mathsf{Scenario} := \mathsf{Reg}_1|^\pi;\ \mathsf{Reg}_2|^\eta;\ \mathsf{ServProv}|^\zeta;\ \mathsf{ServProv}|^\xi$

$\mathsf{Reg}_1 :=$

$$\mathbf{ip}|_\mathbf{u} \to ip|_{idp1} : \text{MS}_{k^-|_{ca}}(i|_u, \mathsf{pk}(k^-|_u), n_c|.); \quad (1)$$
$$\mathbf{ip}|_\mathbf{u} \mapsto ip|_{idp1} : \text{ZK}(k^-|_u; \mathsf{pk}(k^-|_u); \varnothing; \{\mathbf{n}_{\mathbf{z},\mathbf{a}}|., \mathbf{n}_{\mathbf{z},\mathbf{b}}|.\}); \quad (2)$$
$$ip|_{idp1} \to \mathbf{ip}|_\mathbf{u} : \text{MS}_{k^-|_{idp1}}(i|_u, d_1|_u, d_2|_u, d_3|_u, n_a|.) \quad (3)$$

$\mathsf{Reg}_2 :=$

$$\mathbf{ip}|_\mathbf{u} \to ip|_{idp2} : \text{MS}_{k^-|_{ca}}(i|_u, \mathsf{pk}(k^-|_u), n_c|.); \quad (4)$$
$$\mathbf{ip}|_\mathbf{u} \mapsto ip|_{idp2} : \text{ZK}(k^-|_u; \mathsf{pk}(k^-|_u); \varnothing; \{\mathbf{n}_{\mathbf{z},\mathbf{a}}|., \mathbf{n}_{\mathbf{z},\mathbf{b}}|.\}); \quad (5)$$
$$ip|^\eta_{idp2} \to \mathbf{ip}|^\eta_\mathbf{u} : \text{MS}_{k^-|^\eta_{idp2}}(i|^\eta_u, d_5|^\eta_u, d_6|^\eta_u, n_b|^\eta_\cdot) \quad (6)$$

$\mathsf{ServProv} :=$

$$\mathbf{ip}|_\mathbf{u} \to ip|_{sp} : \text{MS}_{k^-|_{ca}}(i|_u, \mathsf{pk}(k^-|_u), n_c|.), \text{MS}_{k^-|_{idp1}}(i|_u, d_1|_u, d_2|_u, d_3|_u, n_a|.), \text{MS}_{k^-|_{idp2}}(i|_u, d_5|_u, d_6|_u, n_b|.); \quad (1)$$
$$\mathbf{ip}|_\mathbf{u} \mapsto ip|_{sp} : \text{ZK}(k^-|_u; \mathsf{pk}(k^-|_u); \varnothing; \{\mathbf{n}_{\mathbf{z},\mathbf{a}}|., \mathbf{n}_{\mathbf{z},\mathbf{b}}|.\}) \quad (2)$$

Fig. 15 Formalisation of smart certificates: initial knowledge and trace

We introduce the abbreviation $\text{MS}_{\mathsf{k}^-}(\mathsf{m}) := \{\mathsf{m}, S_{\mathsf{k}^-}(\mathsf{m})\}$ to denote a message along with its signature, capturing both X.509 certificates [56] and signed SAML assertions [30].

6.3.1 Smart Certificates

Figure 15 displays our formalisation of smart certificates (§5.3.1). In addition to the knowledge from Figure 14, Alice initially knows her public key certificate

$$\text{MS}_{k^-|^\cdot_{ca}}(i|^\cdot_{al}, \mathsf{pk}(k^-|^\cdot_u), n_c|^\cdot)$$

($n_c|^\cdot$ represents additional information in the certificate such as the validity date), and the corresponding private key $k^-|^\cdot_{al}$. The other items of initial knowledge are the contributions $\mathbf{n}_{\mathbf{z},*}|^*_\cdot$ to Alice's proof of knowledge of $k^-|^\cdot_{al}$, and additional information $n_a|^\pi_\cdot$, $n_b|^\eta_\cdot$ put in the attribute certificates issued by *ii* and *is*.

The messages in the traces Reg_1 and Reg_2 correspond to those in Figure 10(a); the messages in the trace $\mathsf{ServProv}$ correspond to those in Figure 10(b). We model the proof that Alice knows the secret key corresponding to her public key as a ZK proof with secret information $k^-|^\pi_u$ and public information $\mathsf{pk}(k^-|^\pi_u)$.

6.3.2 Linking Service Model

Figure 16 displays the formalisation of the linking service model (§5.3.2). 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 $i|^\cdot_{ls}$, $i|^\cdot_{is}$ 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 $i_l|^\pi_u$; the pseudonyms generated by the identity providers are modelled as $i_{i1,ls}|^\pi_u$ and $i_{i2,ls}|^\pi_u$. Alice's authentication at *ii* during service provision is modelled by the fact that *ii* knows the identifiers $i_{ii}|^*_u$, $* \in \{\zeta, \xi\}$.

$\mathcal{C}^0_{ii}$ = (Fig.14) $\cup \{ip|_{ls}, \mathsf{pk}(k^-|_{ls}), i|_{ls},$
$i|_{is}, i_{i1,ls}|^{\pi}_{u}, \mathbf{n}|^{\pi}_{\cdot}, i_{ii}|^{\zeta}_{u}, \mathbf{i_{sess}}|^{\zeta}_{\mathbf{u}}, \mathbf{n}|^{\zeta}_{\cdot}, i_{ii}|^{\xi}_{u}, \mathbf{i_{sess}}|^{\xi}_{\mathbf{u}}, \mathbf{n}|^{\xi}_{\cdot}\}$;
$\mathcal{C}^0_{is}$ = (Fig.14) $\cup \{ip|_{ls}, \mathsf{pk}(k^-|_{ls}), i|_{ls}, i|_{is}, i_{i2,ls}|^{\eta}_{u}, \mathbf{n}|^{\eta}_{\cdot}\}$;
$\mathcal{C}^0_{bs}$ = (Fig.14) $\cup \{ip|_{ls}, \mathsf{pk}(k^-|_{ls}), i|_{ls}, i|_{is}\}$;
$\mathcal{C}^0_{ls}$ = (Fig.14) $\cup \{ip|_{ls}, \mathsf{pk}(k^-|_{ls}), k^-|_{ls}, i|_{ls}, i|_{is}, i_l|^{\nu}_{al}, i_l|^{\pi}_{u}, i_l|^{\eta}_{u}, \mathbf{n'}|^{\zeta}_{\cdot}, \mathbf{n'}|^{\xi}_{\cdot}\}$

$\mathsf{Scenario} := \mathsf{Reg}_1|^{\pi};\ \mathsf{Reg}_2|^{\eta};\ \mathsf{ServProv}|^{\zeta};\ \mathsf{ServProv}|^{\xi}$
$\mathsf{Reg}_1 :=$
$ip|_{idp1} \to ip|_{ls} : \mathrm{MS}_{k^-|_{idp}}(i_{i1,ls}|_u, \mathbf{n}|_{\cdot})$ (1)
$\mathsf{Reg}_2 :=$
$ip|_{idp2} \to ip|_{ls} : \mathrm{MS}_{k^-|_{idp}}(i_{i2,ls}|_u, \mathbf{n}|_{\cdot})$ (2)
$\mathsf{ServProv} :=$
$ip|_{idp1} \to ip|_{sp} : \mathrm{MS}_{k^-|_{idp1}}(\mathbf{i_{sess}}|_{\mathbf{u}}, d_1|_u, d_2|_u, i|_{ls}, E_{\mathsf{pk}(k^-|_{ls})}(i_{i1,ls}|_u, \mathbf{n}|_{\cdot}))$ (1)
$ip|_{sp} \to ip|_{ls} : E_{\mathsf{pk}(k^-|_{ls})}(i_{i1,ls}|_u, \mathbf{n}|_{\cdot}), \mathrm{MS}_{k^-|_{idp1}}(\mathbf{i_{sess}}|_{\mathbf{u}}, d_1|_u, d_2|_u, i|_{ls}, E_{\mathsf{pk}(k^-|_{ls})}(i_{i1,ls}|_u, \mathbf{n}|_{\cdot}))$; (2)
$ip|_{ls} \to ip|_{sp} : i|_{idp2}, E_{\mathsf{pk}(k^-|_{idp2})}(i_{i2,ls}|_u, \mathbf{n'}|_{\cdot})$; (3)
$ip|_{sp} \to ip|_{idp2} : E_{\mathsf{pk}(k^-|_{idp2})}(i_{i2,ls}|_u, \mathbf{n'}|_{\cdot}), \mathrm{MS}_{k^-|_{idp1}}(\mathbf{i_{sess}}|_{\mathbf{u}}, d_1|_u, d_2|_u, i|_{ls}, E_{\mathsf{pk}(k^-|_{ls})}(i_{i1,ls}|_u, \mathbf{n}|_{\cdot}))$; (4)
$ip|_{idp2} \to ip|_{sp} : \mathrm{MS}_{k^-|_{idp2}}(\mathbf{i_{sess}}|_{\mathbf{u}}, d_6|_u)$ (5)

Fig. 16 Formalisation of linking service model: initial knowledge and trace

The registration and service provision phases in the trace correspond to Figures 11(a) and 11(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 $\mathbf{n}|_{\cdot}$, $\mathbf{n'}|_{\cdot}$ to ensure that *bs* cannot link different sessions by comparing them.

The linking service model aims to satisfy a privacy requirement specifically about the linking service, which we call *LS attribute undetectability* (LD). We can express this requirement formally in a similar way to the SID, SPD, and ID requirements: $d_1|^{*}_{*} \notin \mathsf{O}_{ls} \wedge \ldots \wedge d_6|^{*}_{*} \notin \mathsf{O}_{ls}$.

The linking service model in general is independent from message formats. However, the authors also present an instantiation using the SAML 2.0 [30] and Liberty ID-WSF 2.0 [54] standards. Our model captures that instantiation.

6.3.3 Identity Mixer

The formalisation of the scenario when using Identity Mixer (§5.3.3) is shown in Figure 17. The most notable piece of initial knowledge is Alice's secret identifier $i|_{al}$. In the trace, registration follows the steps of Figure 12(a); service provision is as in Figure 12(b). For our purposes, we can represent the commitment to Alice's secret identifier in the first message by a hash $\mathcal{H}(i|^{\pi}_{u}, n_{c1,1}|^{\pi}_{\cdot})$. By inference rule ($\vdash$**EI**$_2$), Alice learns a credential from the issuing protocol linking her attributes to her secret identifier. For instance, from message (2) she can derive

$$\mathrm{cred}^{i|_u}_{k^-|_{idp1}}(i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u; n_{c1,2}|_{\cdot}, n_{c1,5}|_{\cdot}, n_{c1,6}|_{\cdot})|^{\pi}.$$

Note that this credential contains Alice's identifier $i_{ii}|^{\pi}_{u}$ as an additional attribute: it is used later for anonymity revocation.

In the first message of service provision, again we represent the commitments to Alice's secret identifier and at-

$\mathcal{C}^0_{al}$ = (Fig.14) $\cup \{i|_{al},$
$n_{c1,1}|^{\pi}_{\cdot}, n_{c1,2}|^{\pi}_{\cdot}, n_{c1,3}|^{\pi}_{\cdot}, n_{c1,7}|^{\pi}_{\cdot}, n_{c2,1}|^{\eta}_{\cdot}, n_{c2,2}|^{\eta}_{\cdot}, n_{c2,3}|^{\eta}_{\cdot}, n_{c2,7}|^{\eta}_{\cdot},$
$\mathbf{n_v}|^{\zeta}_{\cdot}, cnd|^{\zeta}_{\cdot}, \mathbf{n}|^{\zeta}_{\cdot}, \mathbf{n_{1,1}}|^{\zeta}_{\cdot}, \mathbf{n_{1,2}}|^{\zeta}_{\cdot}, \mathbf{n_{1,3}}|^{\zeta}_{\cdot}, \mathbf{n_{1,a}}|^{\zeta}_{\cdot}, \mathbf{n_{2,1}}|^{\zeta}_{\cdot}, \mathbf{n_{2,a}}|^{\zeta}_{\cdot},$
$\mathbf{n_v}|^{\xi}_{\cdot}, cnd|^{\xi}_{\cdot}, \mathbf{n}|^{\xi}_{\cdot}, \mathbf{n_{1,1}}|^{\xi}_{\cdot}, \mathbf{n_{1,2}}|^{\xi}_{\cdot}, \mathbf{n_{1,3}}|^{\xi}_{\cdot}, \mathbf{n_{1,a}}|^{\xi}_{\cdot}, \mathbf{n_{2,1}}|^{\xi}_{\cdot}, \mathbf{n_{2,a}}|^{\xi}_{\cdot}\}$
$\mathcal{C}^0_{ii}$ = (Fig.14) $\cup \{n_{c1,4}|^{\pi}_{\cdot}, n_{c1,5}|^{\pi}_{\cdot}, n_{c1,6}|^{\pi}_{\cdot}\}$;
$\mathcal{C}^0_{is}$ = (Fig.14) $\cup \{n_{c2,4}|^{\eta}_{\cdot}, n_{c2,5}|^{\eta}_{\cdot}, n_{c2,6}|^{\eta}_{\cdot}\}$;
$\mathcal{C}^0_{bs}$ = (Fig.14) $\cup \{\mathbf{n_{1,b}}|^{\zeta}_{\cdot}, \mathbf{n_{2,b}}|^{\zeta}_{\cdot}, \mathbf{n_{1,b}}|^{\xi}_{\cdot}, \mathbf{n_{2,b}}|^{\xi}_{\cdot}\}$

$\mathsf{Scenario} := \mathsf{Reg}_1|^{\pi};\ \mathsf{Reg}_2|^{\eta};\ \mathsf{ServProv}|^{\zeta};\ \mathsf{ServProv}|^{\xi}$
$\mathsf{Reg}_1 :=$
$\mathbf{ip}|_{\mathbf{u}} \to ip|_{idp1} : \mathcal{H}(i|_u, n_{c1,1}|_{\cdot});$ (1)
$\mathbf{ip}|_{\mathbf{u}} \mapsto ip|_{idp1} : \mathrm{ICred}^{i|_u}_{k^-|_{idp1}}(i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u; \{n_{c1,i}|_{\cdot}\}^7_{i=1})$ (2)
$\mathsf{Reg}_2 :=$
$\mathbf{ip}|_{\mathbf{u}} \to ip|_{idp2} : \mathcal{H}(i|_u, n_{c2,1}|_{\cdot});$ (3)
$\mathbf{ip}|_{\mathbf{u}} \mapsto ip|_{idp2} : \mathrm{ICred}^{i|_u}_{k^-|_{idp2}}(d_5|_u, d_6|_u; \{n_{c2,i}|_{\cdot}\}^7_{i=1})$ (4)
$\mathsf{ServProv} :=$
$\mathbf{ip}|_{\mathbf{u}} \to ip|_{sp} : \mathcal{H}(i|_u, \mathbf{n}|_{\cdot}), \mathcal{H}(i_{ii}|_u, \mathbf{n_{1,2}}|_{\cdot}), \mathcal{H}(d_2|_u, \mathbf{n_{1,1}}|_{\cdot}), \mathcal{H}(d_3|_u, \mathbf{n_{1,3}}|_{\cdot}), d_1|_u, d_2{>}60|_u, cnd|_{\cdot}, \mathsf{pk}(k^-|_{ttp}), E_{\mathsf{pk}(k^-|_{ttp})}(i_{ii}|_u, \mathbf{n_v}|_{\cdot})_{cnd|_{\cdot}};$ (1)
$\mathbf{ip}|_{\mathbf{u}} \mapsto ip|_{sp} : \mathrm{ZK}(\mathrm{cred}^{i|_u}_{k^-|_{idp1}}(i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u; n_{c1,2}|_{\cdot}, n_{c1,5}|_{\cdot}), i|_u, i_{ii}|_u, d_1|_u, d_2|_u, d_3|_u, \mathbf{n}|_{\cdot}, \mathbf{n_{1,2}}|_{\cdot}, \mathbf{n_{1,1}}|_{\cdot}, \mathbf{n_{1,3}}|_{\cdot}; \mathcal{H}(i|_u, \mathbf{n}|_{\cdot}), \mathcal{H}(i_{ii}|_u, \mathbf{n_{1,2}}|_{\cdot}), \mathcal{H}(d_2|_u, \mathbf{n_{1,1}}|_{\cdot}), \mathcal{H}(d_3|_u, \mathbf{n_{1,3}}|_{\cdot}), d_1|_u, \mathsf{pk}(k^-|_{idp1}), \mathsf{pk}(k^-|_{ttp}), E_{\mathsf{pk}(k^-|_{ttp})}(i_{ii}|_u, \mathbf{n_v}|_{\cdot})_{cnd|_{\cdot}}; d_2{>}60|_u; \{\mathbf{n_{1,a}}|_{\cdot}, \mathbf{n_{1,b}}|_{\cdot}\});$ (2)
$\mathbf{ip}|_{\mathbf{u}} \to ip|_{sp} : \mathcal{H}(i|_u, \mathbf{n}|_{\cdot}), \mathcal{H}(d_5|_u, \mathbf{n_{2,1}}|_{\cdot}), d_6|_u, cnd|_{\cdot};$ (3)
$\mathbf{ip}|_{\mathbf{u}} \mapsto ip|_{sp} : \mathrm{ZK}(\mathrm{cred}^{i|_u}_{k^-|_{idp2}}(d_5|_u, d_6|_u; n_{c2,2}|_{\cdot}, n_{c2,5}|_{\cdot}), i|_u, d_5|_u, d_6|_u, \mathbf{n}|_{\cdot}, \mathbf{n_{2,1}}|_{\cdot}; \mathcal{H}(i|_u, \mathbf{n}|_{\cdot}), \mathcal{H}(d_5|_u, \mathbf{n_{2,1}}|_{\cdot}), d_6|_u, \mathsf{pk}(k^-|_{idp2}); \varnothing; \{\mathbf{n_{2,a}}|_{\cdot}, \mathbf{n_{2,b}}|_{\cdot}\})$ (4)

Fig. 17 Formalisation of Identity Mixer: initial knowledge and trace

tributes by hashes. For anonymity revocation purposes, the first message additionally includes an encryption of the identifier $i_{ii}|_u^{\pi}$ for the trusted third party, with a condition $cnd|_.$ attached describing when the anonymity of the transaction may be revoked. The ZK proof in message (2) convinces *bs* that:

- Alice owns a credential, signed with *ii*'s private key;
- the secret identifier and attributes in the credential correspond to the values or commitments sent previously;
- the property $d_2{>}60|_u$ 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 $\mathcal{H}(i|_u, \mathbf{n}|_.)$ in messages (1) and (3) is the same, guaranteeing *bs* that the two certificates are indeed of the same user.

6.3.4 Smartcard Scheme

The Smartcard scheme (§5.3.4) is formalised in Figure 18. 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 $i|_{al}^{\cdot}$, generated on the card, which is 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 13(a); the messages from the service provision part correspond to Figure 13(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 the service provision phase, $\mathbf{q}|_.$ and $\mathbf{dm}|_.$ represent *bs*'s query: what information it needs, and how recent it should be.

Note that in [100], 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 the address provider 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.

6.4 Step 4: Verify Privacy Properties & Analysis of Results

Step 4 of our framework is to verify which requirements are satisfied by the analysed systems. This step is performed

$$\mathcal{C}_{al}^0 = \mathcal{C}_{al}^s \cup \{\mathsf{pk}(k^-|_{ca}^{\cdot}), \mathrm{MS}_{k^-|_{ca}^{\cdot}}(i_c|_.^{\cdot}, \mathsf{pk}(k_c^-|_.^{\cdot}), n_c|_.^{\cdot}), k_c^-|_.^{\cdot}, i|_{al}^{\cdot},$$
$$\mathbf{n_a}|_.^{\pi}, \mathbf{n_a}|_.^{\eta}, \mathbf{n_v}|_.^{\zeta}, \mathbf{i_{sess}}|_{\mathbf{u}}^{\zeta}, \mathbf{n_a}|_.^{\zeta}, \mathbf{n_v}|_.^{\xi}, \mathbf{i_{sess}}|_{\mathbf{u}}^{\xi}, \mathbf{n_a}|_.^{\xi}\};$$
$$\mathcal{C}_{ii}^0 = \mathcal{C}_{ii}^s \cup \{\mathsf{pk}(k^-|_{ca}^{\cdot}), \mathrm{MS}_{k^-|_{ca}^{\cdot}}(i|_{ii}^{\cdot}, \mathsf{pk}(k^-|_{ii}^{\cdot}), n_{ii}|_.^{\cdot}), \mathbf{n_b}|_.^{\pi}\};$$
$$\mathcal{C}_{is}^0 = \mathcal{C}_{is}^s \cup \{\mathsf{pk}(k^-|_{ca}^{\cdot}), \mathrm{MS}_{k^-|_{ca}^{\cdot}}(i|_{is}^{\cdot}, \mathsf{pk}(k^-|_{is}^{\cdot}), n_{is}|_.^{\cdot}), \mathbf{n_b}|_.^{\eta}\};$$
$$\mathcal{C}_{bs}^0 = \mathcal{C}_{bs}^s \cup \{\mathsf{pk}(k^-|_{ca}^{\cdot}), \mathrm{MS}_{k^-|_{ca}^{\cdot}}(i|_{bs}^{\cdot}, \mathsf{pk}(k^-|_{bs}^{\cdot}), n_{bs}|_.^{\cdot}),$$
$$\mathbf{n_b}|_.^{\zeta}, \mathbf{dm}|_.^{\zeta}, \mathbf{q}|_.^{\zeta}, \mathbf{n_b}|_.^{\xi}, \mathbf{dm}|_.^{\xi}, \mathbf{q}|_.^{\xi}\}$$

$\mathsf{Scenario} := \mathsf{Reg}_1|^{\pi};\ \mathsf{Reg}_2|^{\eta};\ \mathsf{ServProv}|^{\zeta};\ \mathsf{ServProv}|^{\xi}$

$\mathsf{Reg}_1 :=$

$$\mathbf{ip}|_{\mathbf{u}} \to ip|_{idp1} : \mathrm{MS}_{k^-|_{ca}}(i_c|_.^{\cdot}, \mathsf{pk}(k_c^-|_.^{\cdot}), n_c|_.^{\cdot}), \mathbf{n_a}|_.^{\cdot}; \quad (1)$$
$$ip|_{idp1} \to \mathbf{ip}|_{\mathbf{u}} : \mathrm{MS}_{k^-|_{ca}}(i|_{idp1}, \mathsf{pk}(k^-|_{idp1}), n_{idp1}|_.^{\cdot}), \mathbf{n_b}|_.^{\cdot}; \quad (2)$$
$$\mathbf{ip}|_{\mathbf{u}} \to ip|_{idp1} : E'_{\mathrm{AKA}(k_c^-|_.^{\cdot};\mathbf{n_a}|_.^{\cdot};k^-|_{idp1}^{\cdot};\mathbf{n_b}|_.^{\cdot})}(\mathcal{H}(i|_u, i|_{idp1})); \quad (3)$$
$$ip|_{idp1} \to \mathbf{ip}|_{\mathbf{u}} : E'_{\mathrm{AKA}(k_c^-|_.^{\cdot};\mathbf{n_a}|_.^{\cdot};k^-|_{idp1}^{\cdot};\mathbf{n_b}|_.^{\cdot})}(d_1|_u, d_2|_u, d_3|_u) \quad (4)$$

$\mathsf{Reg}_2 :=$

$$\mathbf{ip}|_{\mathbf{u}} \to ip|_{idp2} : \mathrm{MS}_{k^-|_{ca}}(i_c|_.^{\cdot}, \mathsf{pk}(k_c^-|_.^{\cdot}), n_c|_.^{\cdot}), \mathbf{n_a}|_.^{\cdot}; \quad (5)$$
$$ip|_{idp2} \to \mathbf{ip}|_{\mathbf{u}} : \mathrm{MS}_{k^-|_{ca}}(i|_{idp2}, \mathsf{pk}(k^-|_{idp2}), n_{idp2}|_.^{\cdot}), \mathbf{n_b}|_.^{\cdot}; \quad (6)$$
$$\mathbf{ip}|_{\mathbf{u}} \to ip|_{idp2} : E'_{\mathrm{AKA}(k_c^-|_.^{\cdot};\mathbf{n_a}|_.^{\cdot};k^-|_{idp2}^{\cdot};\mathbf{n_b}|_.^{\cdot})}(\mathcal{H}(i|_u, i|_{idp2})); \quad (7)$$
$$ip|_{idp2} \to \mathbf{ip}|_{\mathbf{u}} : E'_{\mathrm{AKA}(k_c^-|_.^{\cdot};\mathbf{n_a}|_.^{\cdot};k^-|_{idp2}^{\cdot};\mathbf{n_b}|_.^{\cdot})}(d_5|_u, d_6|_u) \quad (8)$$

$\mathsf{ServProv} :=$

$$\mathbf{ip}|_{\mathbf{u}} \to ip|_{sp} : \mathrm{MS}_{k^-|_{ca}}(i_c|_.^{\cdot}, \mathsf{pk}(k_c^-|_.^{\cdot}), n_c|_.^{\cdot}), \mathbf{n_a}|_.^{\cdot}; \quad (1)$$
$$ip|_{sp} \to \mathbf{ip}|_{\mathbf{u}} : \mathrm{MS}_{k^-|_{ca}}(i|_{sp}, \mathsf{pk}(k^-|_{sp}), n_{sp}|_.^{\cdot}), \mathbf{n_b}|_.^{\cdot}; \quad (2)$$
$$\mathbf{ip}|_{\mathbf{u}} \to ip|_{sp} : E'_{\mathrm{AKA}(k_c^-|_.^{\cdot};\mathbf{n_a}|_.^{\cdot};k^-|_{sp}^{\cdot};\mathbf{n_b}|_.^{\cdot})}(\mathbf{i_{sess}}|_{\mathbf{u}}); \quad (3)$$
$$ip|_{sp} \to \mathbf{ip}|_{\mathbf{u}} : \mathbf{i_{sess}}|_{\mathbf{u}}, E'_{\mathrm{AKA}(k_c^-|_.^{\cdot};\mathbf{n_a}|_.^{\cdot};k^-|_{sp}^{\cdot};\mathbf{n_b}|_.^{\cdot})}(\mathbf{dm}|_.^{\cdot}); \quad (4)$$
$$ip|_{sp} \to \mathbf{ip}|_{\mathbf{u}} : \mathbf{i_{sess}}|_{\mathbf{u}}, E'_{\mathrm{AKA}(k_c^-|_.^{\cdot};\mathbf{n_a}|_.^{\cdot};k^-|_{sp}^{\cdot};\mathbf{n_b}|_.^{\cdot})}(\mathbf{q}|_.^{\cdot}); \quad (5)$$
$$\mathbf{ip}|_{\mathbf{u}} \to ip|_{sp} : E'_{\mathrm{AKA}(k_c^-|_.^{\cdot};\mathbf{n_a}|_.^{\cdot};k^-|_{sp}^{\cdot};\mathbf{n_b}|_.^{\cdot})}(d_1|_u, d_2{>}60|_u, d_6|_u,$$
$$E_{\mathsf{pk}(k^-|_{ttp})}(\mathcal{H}(i|_u, i|_{idp1}), \mathbf{n_v}|_.^{\cdot})) \quad (6)$$

Fig. 18 Formalisation of Smartcard scheme: initial knowledge and trace

automatically using our Prolog tool (§3.3): given the formalised requirements (§6.2) and communication in the systems (§6.3), the tool automatically determines which requirements hold in which systems. (More precisely, it computes the state that the given initial state evolves into by the given trace, also checking trace validity (see Appendix A); then computes the views of actors and coalitions in this state; and finally, verifies which of the given requirements hold in these views.) The results are shown in Table 6: we now analyse them.

6.4.1 Non-privacy requirements

The two non-privacy requirements *attribute exchange (AX)* and *anonymity revocation (AR)* are satisfied in all systems. Indeed, attribute exchange is the basic requirement of an

Scheme	AX	AR	SID	SPD	ID	IM	ISM	SL	IL	IIL	ISL
Smart certificates	✓	✓	✗	✗	✓	✓	✓	✗	✓	✗	✗
Linking service model	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗
Identity Mixer	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓†
Smartcard scheme	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 6 Comparison of privacy requirements 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 3) . †: may not be satisfiable efficiently depending on non-privacy-related requirements.

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 fulfil this requirement.

6.4.2 Detectability requirements

The detectability requirements with respect to the service provider, *property-attribute undetectability (SPD)* and *irrelevant attribute undetectability (SID)*, verify the possibility to reveal properties of 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 requirement. 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 properties of 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 [77]. 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 properties of an attribute instead of its value. Recently, an extension to SAML to achieve this has been proposed [73]. With this extension (or other instantiations), the requirement 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.

6.4.3 Involvement requirements

The involvement requirements 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 [31], 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.

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 [31]: 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 requirement may be to use group signatures [33] 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.

6.4.4 Linkability requirements

Finally, we discuss the results for the linkability requirements. *Session unlinkability (SL)* is a natural requirement 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 requirement 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 [35] can be used so that at least the non-authenticating identity provider does not learn which user he is providing attributes of.

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 requirement. 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 does not fix IIL in our model.

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 [23]. In both cases, revocation is not immediate.

For Identity Mixer, two proposals for immediate revocation have been done [27]. 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 [23]. 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 [23] this may still not be efficient enough. Note that the Smartcard scheme does not support immediate revocation at all.

7 Discussion

In this section, we discuss several applicability aspects of our analysis framework: what privacy requirements can be verified, how the scenario should be defined, and what systems can be modelled. We also discuss possible generalisations, and effort needed to analyse a new system.

Privacy Requirements Our framework can be used to verify any data minimisation requirement expressible in terms of the elementary detectability, linkability, and involvement requirements described in Section 2.3. Although the case study demonstrates that this includes many relevant requirements proposed in the literature, there are also privacy aspects that our model does not capture. Most significantly, we allow only limited reasoning (via attribute properties) about the meaning of pieces of personal information other than identifiers. For instance, we do not allow a piece of information to be inferred from several others, e.g. "address" follows from "street name" and "house number". Also, we do not consider (probabilistic) links due to combinations of non-identifying attributes, e.g., matching name and post code from two profiles imply a link with high probability. This

choice reflects the goal of our approach, namely to compare the relative privacy of different systems (that differ in what identifiers are used and how). 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 (see Section 8).

Apart from explicitly transferred information, i.e., the user's attributes, we analyse one particular kind of implicitly transferred information; namely, involvement requirements. 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 (see, e.g., privacy issues in smart metering systems [82]). Knowledge about numbers of transactions can be expressed in our model; date and time may be appended as "tags" to communication.

Scenario-Dependence Our analysis framework requires the specification of a scenario. In particular, this scenario needs to be designed in such a way that all privacy properties to be verified can be phrased in terms of personal information occuring in the scenario. It is straightforward to analyse variants of the scenario by modifying it, but this does involve some work in practice. Our analysis framework and its implementation are designed to verify properties of particular elements in a particular trace; both need to be modified for other scenarios. This task can be lightened by exploiting Prolog's programming features. For instance, the scenario in our case study involves two traces of service provisions, which are almost the same; in our implementation of the model of the systems, both are generated by one Prolog predicate which takes the variable elements as input. This approach can also be used to generate traces with more actors or protocols, and to generate lists of checks that need to be performed for a given privacy requirement. Since the conclusions of an analysis depend on the scenario, it should be chosen carefully to capture all relevant privacy aspects. We refer to [98] for a symbolic extension of our framework which is independent from a particular scenario.

Adaptation and Generalisation Our framework is designed to be general enough for the analysis of any system in which actors use communication protocols to exchange personal information. If the message format of the communication protocols in the system is available, then the main difficulty in modelling the system is to make sure the cryptographic primitives used in the protocols are accurately modelled. Although the present work models several frequently-used primitives, the model may need to be adapted to reflect characteristics of the particular implementations used (e.g., digital signatures may be with message recovery instead of with appendix, meaning that the message can be derived from the signature [69]); or new cryptographic primitives may need to be added. Once this is done, modelling the actual protocols is usually a matter of industrious bookkeeping.

To give the reader a flavour of the effort needed to model new primitives, we draw upon our experiences in extending the basic formal model of [99] to perform the case study in this paper. Some operations are easily expressible in terms of standard primitives. For instance, for our purposes, commitments can be modelled as hashes because they satisfy the same inference rules. When modelling primitives, it is helpful to look at existing formalisations, e.g. using deductive systems [37,49] or equational theories [2,16]: they can usually be translated to the three-layer model. For instance, the formalisation of labelled encryption used in this work is based on [26]. Special attention should be paid to testing rules. Deductive systems do not usually consider testing; equational theories can include rules, e.g., for signature verification (e.g., [43]), which translate to testing rules in the three-layer model, but may include only those rules that were relevant to the analysis at hand. Thus, to obtain a complete set of testing rules, one needs to take a lower-level look at the operation of the primitive. In addition, note that existing formalisations (e.g., [26]) may not explicitly model randomness in non-deterministic primitives; however, in our model this is needed because we assume messages to be deterministic.

In some cases, no suitable existing formalisation of a cryptographic primitive may be available. In such a case, the general (security) definition of the primitive (e.g., [41] for ZK proofs) generally suffices for obtaining a description for the language $\mathcal{L}^c$. However, different implementations of a primitive may give rise to different inference rules. Thus, to obtain inference rules, one needs to consider the particular implementation used in the protocol under analysis. In our experience, this is feasible. Note that because we are only interested in privacy aspects of the primitives, usually some simplifications can be made. See Appendix B for two examples: ZK proofs and anonymous credentials.

As mentioned in Section 3, our model imposes several assumptions on the cryptographic primitives and operations modelled. In particular, because we assume that differently-constructed messages cannot have coinciding contents, we cannot model some operations such as "exclusive or" (which satisfies that $x \oplus (x \oplus y) = y$). Also, our visible failure assumption may cause an over-approximation the knowledge of actors: in our model, actors can draw conclusions from the fact that a cryptographic operation was applied successfully, in practice, this may not be possible. These limitations may be overcome by generalising our model through its connection with static equivalence (see Section 8); we leave this as future work.

8 Related Work

We discuss related work on our privacy analysis framework (§8.1), and on the identity management case study (§8.2).

8.1 Privacy Analysis Framework

The analysis of privacy entails two orthogonal concerns: what information is leaked by how identifiers and other pieces of information are exchanged in communication protocols; and what inferences can be made from the information learned in this way. The present work addresses the former concern, which we discuss first; afterwards, we briefly discuss the latter concern.

Formal analysis techniques have been applied to communication protocols for many years, mainly to verify security properties [2,22,68,78]. Most formal methods rely on two basic ideas: the Dolev-Yao attacker model and state exploration techniques. In the Dolev-Yao attacker model, one considers communication messages using idealised cryptographic primitives, and an attacker who controls some or all communication channels between legitimate parties (meaning that he can insert and suppress messages at will, and fabricate messages based on his observations). The reasoning that the attacker performs to fabricate messages can be described by deductive systems (e.g., [37,49]) or equational theories (e.g., [2,16]). State space exploration techniques assess the system security by analysing all possible evolutions of a given system in the presence of a Dolev-Yao attacker. The requirements of a system are then verified by checking whether any of the states that can be reached by the system correspond to an attack (e.g., the attacker knows a secret, or has succeeded in impersonating a legitimate user). Several process algebras [2,19,70] provide machinery to perform state space exploration. Other approaches have also been proposed, e.g., using induction [78].

Recently, more and more work has focused on the use of these techniques for privacy properties, in application domains such as electronic toll collection [42], e-voting [43, 45], RFID systems [21], and Direct Anonymous Attestation [88]. These proposals express privacy in terms of "experiments": slightly different settings for the execution of the same protocol that should be indistinguishable to an attacker. For instance, in electronic toll collection, an attacker should not be able to distinguish a setting in which a first car takes a left road and a second car takes a right road from a situation in which the first car takes the right road and the second car takes the left road. Similarly, in Direct Anonymous Attestation, an attacker should not be able to distinguish a signature produced by one trusted platform module from a signature produced by another one.

Conceptually, our work differs from these existing formal methods in several ways. We provide general definitions for detectability and associability that take into account different data subjects that may occur in a single protocol instance; conversely, existing works either provide specific definitions tailored to a particular setting or protocol [42, 43,45], or only consider links between messages and their senders [5]. Moreover, we explicitly model the knowledge of (coalitions of) legitimate actors in the system as needed for analysis of data minimisation, whereas existing methods focus on (malicious) outsiders. Also, we consider knowledge in a particular scenario, whereas existing methods focus on a family of scenarios. Although particular queries in our analysis framework could be translated to queries using these existing formal methods (e.g., using frameworks like [34] to convert a trace to a set of actions by protocol roles), we expect that it is infeasible to design a completely automatic translation to queries that the tools available today are able to handle. Conversely, our privacy analysis framework achieves practical privacy analysis and comes with an implementation.

At a technical level, however, there are similarities between our model for reasoning about knowledge and existing models. Existing models reason about knowledge of an attacker about message contents. Three popular definitions cover whether an attacker knows the contents of a given piece of information: weak secrecy [15], resistance against guessing attacks [39], and strong secrecy [15]. The weakest definition, weak secrecy, defines secrecy as non-derivability using a contents-layer deductive system; as shown in [99], this property holds exactly if no context-layer representation of the contents can be derived using our three-layer model.

The other two existing definitions strengthen the concept of weak secrecy by employing the notion of static equivalence [2] of frames. A frame captures the knowledge of an actor at a certain point in time. Intuitively, two frames are statically equivalent if an actor cannot distinguish between the situations modelled by the two frames. Resistance against guessing attacks [39] of a frame models that an actor should not have any way to verify if a guess he makes about the contents of a particular piece of information is correct. This is formalised by adding the actor's guess to the frame, and verifying that the situation in which the guess is correct is statically equivalent to the situation in which the guess is incorrect. Intuitively, in our model, the contents c of a piece of information is resistant to guessing attacks if and only if there is no context-layer item with contents c that is known to be content equivalent to a guess with contents c. This link can be made more precise, and can be used to generalise the approach presented in this paper (see [95] for details). One strong point of static equivalence is that it can be formally linked to computational models of cryptography [9]; compared to the equational theory of [9], our visible failure assumption on deterministic symmetric encryption is an over-approximation of knowledge.

Strong secrecy [15] additionally takes into account that the secret may have the same contents as any arbitrary other message, as well as that the value of the secret may influence the behaviour of actors. Our model (as well as the definition of resistance against guessing attacks) does not take these aspects into account, so strong secrecy is, formally speaking, stronger. Strong and weak secrecy are known to coincide [40] under certain conditions in a certain equational theory; an interesting direction for future work is to verify if similar results hold for the equational theories corresponding to our model. We remark that in practice, tools verify an over-approximation of strong secrecy [16] and hence may give false positives.

Similarly, existing notions of linkability [5,42,43,45] are formally based on static equivalences. For instance, in the electronic toll collection example given above, consider any frame corresponding to a system evolution in which a first car with identifier A goes left and a second car with identifier B goes right. Unlinkability means that this frame should be statically equivalent to a frame corresponding to a system evolution when the first car has identifier B and the second car has identifier A. In many cases, corresponding frames differ only by the use of identifiers, in which case static equivalence corresponds to the non-knowledge of content equivalence of these identifiers, like our definition of associability. However, linkability also allows other correspondences and takes into account that the value of the identifier may influence the behaviour of actors, and is thus, formally speaking, more powerful. Also in this case, existing tools over-approximate linkability [16]; in practice, it is difficult to avoid false positives.

There are also technical similarities between our model of particular cryptographic primitives and other models from the literature. Labelled encryption is a straightforward extension of normal encryption; our model is similar to the one in [26]. The internals of (incorrect) protocols for authenticated key agreement have over the years proven a popular target for analysis using formal methods [22,66,78]; however, we have not found prior works that formally model the external behaviour of (correct) authenticated key exchange protocols in a larger system.

For ZK proofs, both high-level and low-level formalisations exist. In [65], a low-level model of the operation of ZK proofs is given; however, it cannot be used for knowledge derivation; also, questions have been raised about its technical correctness [26]. Two high-level formalisations of ZK proofs have been proposed [7,26] that, as ours, allow proofs of a restricted set of properties. The equational theory in [7] models the verification of ZK proofs (as our testing rules); the model of [26] only allows correct ZK proofs to take place and does not express their verification. The latter simplification is not suitable for our method, because verification expresses that an actor learns information in new contexts. Note that both model "signature proofs of knowledge" rather than Σ-proofs; however, our methods can also capture that variant.

Three recent proposals [26,65,88] are relevant for our formal model of anonymous credentials. [65] only considers operational aspects of anonymous credentials. [26] models 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 in our low-level formalisation (Appendix B.2). The showing protocol is formalised in terms of ZK proofs. However, credential issuing is not considered in [26]. Finally, Smyth et al. [88] model joining and signing protocols for ECC-based Direct Anonymous Attestation, which are very similar to issuing and showing protocols for BM-CL-based anonymous credentials [25]. Although our model is based on a different signature scheme [24] and specified at a higher level, their model of signatures generally corresponds to our model of signatures from committed messages in Appendix B.2.

Apart from the concern of learning information leaked by communication protocols, the orthogonal concern of inferences made on learned information has also received substantial attention. In particular, the inference of links based on non-identifiers has been approached from two directions: experimentally linking given data, or theoretically guaranteeing that such linking is impossible. Methods to link data from two databases using non-identifiers have been investigated since the seminal paper of Fellegi and Sunter [48]; see [60] for a recent comparative study of available implementations. Data from more than two sources can be grouped together based on pairwise decisions using domain-dependent [13,76,71,84] or domain-independent [14,32] algorithms, or statistical techniques [83]. On the other hands, statistical frameworks to guarantee that linking personal information in a disclosure to other data is impossible (i.e., anonymity) include k-anonymity [36], ℓ-diversity [67], t-closeness [64] and differential privacy [46]. Koot [59] reports on experiments in which the actual degree of anonymity of particular disclosures is computed. Inferences of attribute values based on other attributes is covered in [81]. Our approach can be complemented with these techniques to obtain a full understanding of privacy leakage due to communication.

8.2 Privacy in Identity Management

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 [28]. Hansen et al. [53] 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 [79] define privacy-enhancing

identity management as preserving the unlinkability between user profiles. Finally, in a general survey, Alpár et al. [3] 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 requirements: linkability, involvement and detectability, respectively. In contrast to the vision of minimising actor knowledge, Landau and Moore argue that preventing service providers from collecting transaction data may not be desirable because it prevents the adoption of IdM systems in practice [62]. This falls into a broader discussion on incentives of participants in IdM systems [4,29] that is out of scope for this work.

This work aims to improve the way privacy by data minimisation is assessed compared to existing comparisons [1, 55]. Both comparisons of IdM systems that we are aware of consider data minimisation as one aspect of a much more general comparison of IdM systems. Data minimisation requirements are specified in a high-level way, and verified manually by inspecting the user interface and documentation of the systems. For instance, [1] 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). [55] considers 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 6.4. However, note that our method can only be used to assess data minimisation requirements *given* what information should be exchanged; to verify if this exchange of information is really needed, or consented to by the user, other methods (e.g., [38]) should be used. Some other aspects of the privacy assessment in [1,55] seem less suitable for formal verification, e.g. the user-friendliness and the use of standards in the systems.

Some formal works on privacy in identity management are available. In [79], privacy-enhancing identity management is defined as preserving unlinkability between different user profiles, and the meaning of linkability and its relationship with related concepts is explored in a semi-formal way. Their informal definitions formed the basis of our original work [96] on representing knowledge of personal information. 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, unlinkability [65,91] and undetectability [26] properties have been considered for Identity Mixer and related anonymous credential schemes. For SAML [30], a standard for the exchange of identity information between identity and service providers used in the linking service model, secrecy properties have been considered [6]. 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, and which cannot.

In this work, we focus on minimising knowledge of personal information by technical means; other works address other aspects of privacy. Landau et al. [61] argue that privacy protection can be achieved not just technically, but also by legal and policy means. Hansen et al. [53] 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. [12] 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 of IdM system impacts privacy. However, interestingly, recent research in behavioural economics suggests that offering transparency to users might actually reduce their privacy by inducing them to release more information [20].

9 Conclusions & Future Work

In this work, we have presented a general formal framework to compare communication protocols with respect to privacy by data minimisation. Requirements relevant in a given setting are formalised independently from any particular communication protocol in terms of the knowledge of (coalitions of) actors in a three-layer model of personal information. These requirements are then verified automatically for particular protocols by computing this knowledge from a description of their communication. Using this formal approach, we obtain results that are precise and verifiable, yet provide enough detail to obtain real insight into privacy differences. In contrast to existing methods, our framework allows for the automated verification of a wide range of privacy requirements in one single model.

Our framework may be generalised and extended along several directions. First, the model of personal information can be made more expressive. For instance, to analyse privacy in application domains where the number and timing of transactions is relevant, the model can be extended to take these aspects into account. Other relevant extensions include pieces of information that refer to multiple data subjects (see [98]); or more flexible reasoning about attribute

properties. Second, the model of cryptographic primitives can be made more general. Our current model is based on two assumptions (structural equivalence and visible failure) that limit the number of cryptographic primitives that can be modelled. We are exploring how these limitations can be overcome by modelling cryptographic primitives using an equational theory. Finally, our model depends on the choice of a particular scenario in which requirements are verified; we refer to [98] for a generalisation of our model that is independent from a particular scenario.

We have demonstrated our framework by performing a privacy comparison of identity management systems. In the process, we have defined a comprehensive and detailed set of privacy requirements; to the best of our knowledge, no such set of requirements was available before. We have modelled 4 representative IdM systems, and verified which of the 11 requirements hold for which systems, giving 44 checks in total. It is worth noting that only 17 of the 44 checks are mentioned as (parts of) requirements in the design of the respective IdM systems. In one instance, we found such a requirement not to hold (a problem which is also mentioned by the authors of the system themselves). In another instance, we clarified the exact setting in which a requirement holds, which may a solution that is unrealistic for performance or accountability reasons. The remaining 27 of the 44 checks do not correspond to requirements explicitly stated by the designers of the IdM systems. In this work, we have established whether they hold or not, leading to a more comprehensive analysis and comparison of IdM systems. Interesting extensions to the case study would be to consider requirements for IdM systems based on the extensions mentioned above (e.g. requirements on knowledge about the number of transactions); and additional IdM systems like U-Prove [75] and the STORK Platform (`https://www.eid-stork.eu/`) as well as other variants of the systems we considered.

Acknowledgements We thank the anonymous reviewers for their useful comments. We thank Berry Schoenmakers for useful technical feedback. This work is partially supported by STW through project 'Identity Management on Mobile Devices' (10522).

References

1. Identity Management Systems (IMS): Identification and Comparison Study. Tech. rep., Independent Centre for Privacy Protection Schleswig-Holstein (2003)
2. Abadi, M., Fournet, C.: Mobile Values, New Names, and Secure Communication. In: Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of programming languages (POPL '01), pp. 104–115. ACM (2001)
3. Alpár, G., Hoepman, J.H., Siljee, J.: The Identity Crisis: Security, Privacy and Usability Issues in Identity Management. eprint CoRR cs.CR:1101.0427 (January 2011)
4. Anderson, R.: Can We Fix the Security Economics of Federated Authentication? In: Proceedings of the 19th International Workshop on Security Protocols (SPW '11), LNCS 7114, pp. 25–32. Springer (2011)
5. Arapinis, M., Chothia, T., Ritter, E., Ryan, M.: Analysing Unlinkability and Anonymity Using the Applied Pi Calculus. In: Proceedings of the 2010 23rd IEEE Computer Security Foundations Symposium (CSF '10), pp. 107–121. IEEE (2010)
6. Armando, A., Carbone, R., Compagna, L., Cuellar, J., Abad, L.T.: 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), pp. 1–10. ACM (2008)
7. Backes, M., Maffei, M., Unruh, D.: 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), pp. 202–215. ACM (2008)
8. Bangerter, E., Camenisch, J., Lysyanskaya, A.: A Cryptographic Framework for the Controlled Release of Certified Data. In: Proceedings of the 12th International Workshop on Security Protocols (SPW '04), LNCS 3957, pp. 20–42. Springer (2004)
9. Baudet, M., Warinschi, B., Abadi, M.: Guessing attacks and the computational soundness of static equivalence. J. Comput. Secur. **18**(5), 909–968 (2010)
10. Bella, G., Paulson, L.: Kerberos Version IV: Inductive Analysis of the Secrecy Goals. In: Proceedings of the 5th European Symposium on Research in Computer Security (ESORICS '98), LNCS 1485, pp. 361–375. Springer (1998)
11. Bhargav-Spantzel, A., Camenisch, J., Gross, T., Sommer, D.: User Centricity: A Taxonomy and Open Issues. J. Comput. Secur. **15**(5), 493–527 (2007)
12. Bhargav-Spantzel, A., Squicciarini, A.C., Young, M., Bertino, E.: Privacy Requirements in Identity Management Solutions. In: Proceedings of the IEEE International Workshop on Human Computer Interaction 2007 (HCI '07), LNCS 4558, pp. 694–702. Springer (2007)
13. Bhattacharya, I., Getoor, L.: Collective entity resolution in relational data. ACM Trans Knowl Discov Data **1**(1) (2007)
14. Bilenko, M., Basu, S., Sahami, M.: 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), pp. 58–65. IEEE (2005)
15. Blanchet, B.: Automatic Proof of Strong Secrecy for Security Protocols. In: Proceedings of the 2004 IEEE Symposium on Security and Privacy (S&P '04), pp. 86–100. IEEE (2004)
16. Blanchet, B., Abadi, M., Fournet, C.: Automated Verification of Selected Equivalences for Security Protocols. J. Log. Algebr. Program. **75**(1), 3–51 (2008)
17. Blanchet, B., Smyth, B., Cheval, V.: ProVerif 1.87beta6: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial (2013). Originally appeared as Bruno Blanchet & Ben Smyth (2011) ProVerif 1.85: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial.
18. Boneh, D.: Twenty Years of Attacks on the RSA Cryptosystem. Not. Am. Math. Soc. **46**(2), 1–16 (1999)
19. Boreale, M.: Symbolic Trace Analysis of Cryptographic Protocols. In: Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP '01), LNCS 2076, pp. 667–681. Springer (2001)
20. Brandimarte, L., Acquisti, A., Loewenstein, G.: Misplaced Confidences: Privacy and the Control Paradox. In: Ninth Workshop on the Economics of Information Security (WEIS '10) (2010)
21. Brusó, M., Chatzikokolakis, K., den Hartog, J.: Formal Verification of Privacy for RFID Systems. In: Proceedings of the 23rd IEEE Computer Security Foundations Symposium (CSF '10), pp. 75–88. IEEE (2010)

22. Burrows, M., Abadi, M., Needham, R.: A Logic of Authentication. ACM Trans. Comput. Syst. **8**, 18–36 (1990)
23. Camenisch, J., Kohlweiss, M., Soriente, C.: 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, pp. 481–500. Springer (2009)
24. Camenisch, J., Lysyanskaya, A.: A Signature Scheme with Efficient Protocols. In: Proceedings of the 3rd International Conference on Security in Communication Networks (SCN '02), LNCS 2576, pp. 268–289. Springer (2003)
25. Camenisch, J., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Proceedings of the 24rd Annual International Cryptology Conference (CRYPTO '04), LNCS 3152, pp. 56–72. Springer (2004)
26. Camenisch, J., Mödersheim, S., Sommer, D.: A Formal Model of Identity Mixer. In: Proceedings of the 15th International Workshop on Formal Methods for Industrial Critical Systems (FMICS'10), LNCS 6371, pp. 198–214. Springer (2010)
27. Camenisch, J., Sommer, D., Zimmermann, R.: 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, pp. 25–37. Springer (2006)
28. Cameron, K.: The Laws of Identity. `http://www.identityblog.com/?p=352` (2006)
29. Camp, J.: Identity Management's Misaligned Incentives. IEEE Secur. Priv. **8**(6), 90–94 (2010)
30. Cantor, S., Kemp, K., Philpott, R., Maler, E. (eds.): Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. `http://saml.xml.org/saml-specifications`. OASIS Standard, 15 March 2005
31. Chadwick, D., Inman, G.: Attribute Aggregation in Federated Identity Management. IEEE Comput. **42**(5), 33–40 (2009)
32. Chaudhuri, S., Ganti, V., Motwani, R.: Robust Identification of Fuzzy Duplicates. In: Proceedings of the 21st International Conference on Data Engineering, pp. 865–876. IEEE (2005). DOI 10.1109/ICDE.2005.125
33. Chaum, D., van Heyst, E.: Group Signatures. In: Proceedings of EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques (EUROCRYPT '91), LNCS 547, pp. 257–265. Springer (1991)
34. Chevalier, Y., Rusinowitch, M.: Compiling and securing cryptographic protocols. Inf. Process. Lett. **110**(3), 116–122 (2010)
35. Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private Information Retrieval. In: Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS '95), pp. 41–50. IEEE (1995)
36. Ciriani, V., de Capitani di Vimercati, S., Foresti, S., Samarati, P.: *k*-Anonymity. In: Secure Data Management in Decentralized Systems, AIS 33, pp. 323–353. Springer (2007)
37. Clarke, E.M., Jha, S., Marrero, W.R.: 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), pp. 87–106. Chapman & Hall, Ltd. (1998)
38. Compagna, L., Khoury, P.E., Krausová, A., Massacci, F., Zannone, N.: How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns. Artif. Intell. Law **17**(1), 1–30 (2009)
39. Corin, R., Doumen, J., Etalle, S.: Analysing Password Protocol Security Against Off-line Dictionary Attacks. Electron. Notes Theor. Comput. Sci. **121**, 47–63 (2005)
40. Cortier, V., Rusinowitch, M., Zălinescu, E.: Relating Two Standard Notions of Secrecy. In: Proceedings of the 20th International Workshop on Computer Science Logic (CSL '06), LNCS 4207, pp. 303–318. Springer (2006)
41. Cramer, R.: Modular Design of Secure yet Practical Cryptographic Protocols. Ph.D. thesis, Universiteit van Amsterdam (1997)
42. Dahl, M., Delaune, S., Steel, G.: 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, pp. 98–112. Springer (2011)
43. Delaune, S., Kremer, S., Ryan, M.: Verifying privacy-type properties of electronic voting protocols. Comput. Secur. **17**(4), 435–487 (2009)
44. Dolev, D., Yao, A.C.: On the security of public key protocols. Foundations of Computer Science, Annual IEEE Symposium on **0**, 350–357 (1981)
45. Dreier, J., Lafourcade, P., Lakhnech, Y.: A Formal Taxonomy of Privacy in Voting Protocols. Tech. rep., Verimag (2011)
46. Dwork, C.: Differential Privacy. In: Proceedings of 33rd International Colloquium on Automata, Languages and Programming (ICALP '06), LNCS 4052, pp. 1–12. Springer (2006)
47. Erdos, M., Cantor, S.: The Shibboleth Architecture. Tech. rep., Internet2 Consortium (2005). Internet2-mace-shibboleth-arch-protocols-200509
48. Fellegi, I.P., Sunter, A.B.: A Theory for Record Linkage. J Am Stat Assoc **64**(328), 1183–1210 (1969)
49. Fiore, M., Abadi, M.: Computing Symbolic Models for Verifying Cryptographic Protocols. In: Proceedings of the 14th IEEE workshop on Computer Security Foundations (CSFW '01), pp. 160–173. IEEE (2001)
50. Fujioka, A., Okamoto, T., Ohta, K.: A Practical Secret Voting Scheme for Large Scale Elections. In: Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques (AUSCRYPT '92), LNCS 718, pp. 244–251. Springer (1993)
51. Fujisaki, E., Okamoto, T.: Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. In: Proceedings of the 17th Annual International Cryptology Conference (CRYPTO'97), LNCS 1294, pp. 16–30. Springer (1997)
52. Fyffe, G.: Addressing the insider threat. Netw. Secur. **2008**(3), 11–14 (2008)
53. Hansen, M., Berlich, P., Camenisch, J., Clauß, S., Pfitzmann, A., Waidner, M.: Privacy-Enhancing Identity Management. Inf. Secur. Tech. Rep. **9**(1), 35–44 (2004)
54. Hodges, J., Kemp, K., Aarts, R., Whitehead, G., (eds.), P.M.: Liberty ID-WSF SOAP Binding Specification. `http://projectliberty.org/`. Version 2.0
55. Hoepman, J.H., Joosten, R., Siljee, J.: 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, AICT 298, pp. 184–196. Springer (2008)
56. Housley, R., Polk, W., Ford, W., Solo, D.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280
57. Jøsang, A., Pope, S.: User-Centric Identity Management. In: Proceedings of AusCERT Asia Pacific Information Technology Security Conference (AusCERT '05). University of Queensland (2005)
58. Kellomäki, S.e.: TAS^3 Architecture. Tech. Rep. Deliveral D2.1, work package WP2, version 17, TAS^3 project (2009)
59. Koot, M.R.: Measuring and predicting anonymity. Ph.D. thesis, University of Amsterdam (2012)
60. Köpcke, H., Rahm, E.: Frameworks for entity matching: A comparison. Data Knowl Eng **69**(2), 197–210 (2010)
61. Landau, S., Gong, H., Wilton, R.: 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, pp. 51–70. Springer (2009)
62. Landau, S., Moore, T.: Economic Tussles in Federated Identity Management. In: Tenth Workshop on the Economics of Information Security (WEIS '11) (2011)

63. Law, L., Menezes, A., Qu, M., Solinas, J., Vanstone, S.: An Efficient Protocol for Authenticated Key Agreement. Designs, Codes and Cryptography **28**, 119–134 (2003)
64. Li, N., Li, T., Venkatasubramanian, S.: t-Closeness: Privacy Beyond k-Anonymity and ℓ-Diversity. In: Proceedings of International Conference on Data Engineering (ICDE '07), pp. 106–115. IEEE (2007)
65. Li, X., Zhang, Y., Deng, Y.: Verifying Anonymous Credential Systems in Applied Pi Calculus. In: Proceedings of the 8th International Conference on Cryptology and Network Security (CANS '09), LNCS 5888, pp. 209–225. Springer (2009)
66. Lowe, G.: 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, pp. 147–166. Springer (1996)
67. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: ℓ-diversity: Privacy beyond k-anonymity. ACM Trans Knowl Discov Data **1**(1) (2007)
68. Meadows, C.: Formal Methods for Cryptographic Protocol Analysis: Emerging Issues and Trends. IEEE Sel. Areas Commun. **21**(1), 44–54 (2003)
69. Menezes, A.J., Vanstone, S.A., Oorschot, P.C.v.: Handbook of Applied Cryptography, 1st edn. CRC Press, Inc. (1996)
70. Milner, R.: Communicating and Mobile Systems: the π-Calculus. Cambridge University Press (1999)
71. Mray, N., Reitsma, J., Ravelli, A., Bonsel, G.: Probabilistic record linkage is a valid and transparent tool to combine databases without a patient identification number. J Clin Epidemiol **60**(9), 883–891 (2007)
72. Nanda, A.: A Technical Reference for the Information Card Profile V1.0. `http://msdn.microsoft.com/en-us/library/bb298802.aspx` (2007)
73. Neven, G., Preiss, F.S.: Attribute Predicate Profile of SAML and XACML. XACML mailing list, Mar 23 2011 (`http://markmail.org/message/2dha2sqmgni7wpc5`)
74. Data protection guidelines on research in the health sector. Office of the Data Protection Commissioner of Ireland (2007)
75. Paquin, C., Thompson, G.: U-Prove CTP White Paper. Tech. rep., Microsoft (2010)
76. Parag, Domingos, P.: Multi-relational record linkage. In: Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining (MRDM '04), pp. 31–48. ACM (2004)
77. Park, J.S., Sandhu, R.: Smart Certificates: Extending X.509 for Secure Attribute Services on the Web. In: Proceedings of the 22nd National Information Systems Security Conference (NISSC '99), pp. 337–348. US Government Printing Office (1999)
78. Paulson, L.C.: The Inductive Approach to Verifying Cryptographic Protocols. Comput. Secur. **6**(1-2), 85–128 (1998)
79. Pfitzmann, A., Hansen, M.: 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`. V0.32
80. 2011 Cost of Data Breach Study: Global. Ponemon Institute (2011)
81. Pontes, T., Magno, G., Vasconcelos, M.A., Gupta, A., Almeida, J.M., Kumaraguru, P., Almeida, V.: Beware of What You Share: Inferring Home Location in Social Networks. In: 12th International Conference on Data Mining Workshops (ICDMW '12), pp. 571–578. IEEE (2012)
82. Rial, A., Danezis, G.: Privacy-Preserving Smart Metering. In: Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society (WPES '11), pp. 49–60. ACM (2011)
83. Sadinle, M., Fienberg, S.E.: A Generalized Fellegi-Sunter Framework for Multiple Record Linkage With Application to Homicide Record Systems. arXiv 1205.3217 (2012)
84. Sapena, E., Padró, L., Turmo, J.: 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, pp. 428–439. Springer (2008)
85. Schnorr, C.P.: Efficient Identification and Signatures for Smart Cards. In: Proceedings of the 9th Annual International Cryptology Conference (CRYPTO '89), LNCS 434, pp. 239–252. Springer (1989)
86. Seamons, K., Winslett, M., Yu, T., Yu, L., Jarvis, R.: Protecting Privacy during On-Line Trust Negotiation. In: Proceedings of the 2nd International Conference on Privacy Enhancing Technologies (PET '02), LNCS 2482, pp. 249–253. Springer (2003)
87. Smedinghoff, T.J.: Federated Identity Management: Balancing Privacy Rights, Liability Risks, and the Duty to Authenticate. SSRN eLibrary (2009)
88. Smyth, B., Ryan, M., Chen, L.: Formal analysis of anonymity in Direct Anonymous Attestation schemes. In: Proceedings of the 8th International Workshop on Formal Aspects of Security & Trust (FAST '10), LNCS 7140, pp. 245–262. Springer (2011)
89. Sommer, D., Mont, M.C., Pearson, S.: PRIME Architecture V3. Tech. rep., PRIME consortium (2008). Version 1.0
90. Spiekermann, S., Cranor, L.F.: Engineering Privacy. IEEE Trans. Software Eng. **35**(1), 67–82 (2009)
91. Suriadi, S.: Strengthening and formally verifying privacy in identity management systems. Ph.D. thesis, Queensland University of Technology (2010)
92. Tinabo, R., Mtenzi, F., O'Shea, B.: Anonymisation vs. Pseudonymisation: Which one is most useful for both privacy protection and usefulness of e-healthcare data. In: Proceedings of the International Conference on Internet Technology and Secured Transactions (ICITST '09), pp. 1–6. IEEE (2009)
93. Troncoso, C.: Design and analysis methods for privacy technologies. Ph.D. thesis, Katholieke Universiteit Leuven (2011)
94. Prescription drug data: HHS has issued health privacy and security regulations but needs to improve guidance and oversight. U.S. Govt. Accountability Office (2012)
95. Veeningen, M.: Objective Privacy. Ph.D. thesis, Eindhoven University of Technology (2014). (to be submitted)
96. Veeningen, M., de Weger, B., Zannone, N.: Modeling identity-related properties and their privacy strength. In: Proceedings of the 7th International Workshop on Formal Aspects of Security & Trust (FAST '10), LNCS 7140, pp. 126–140. Springer (2011)
97. Veeningen, M., de Weger, B., Zannone, N.: 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, pp. 145–160. Springer (2012)
98. Veeningen, M., Weger, B., Zannone, N.: Symbolic Privacy Analysis through Linkability and Detectability. In: Proceedings of the 7th IFIP WG 11.11 International Conference on Trust Management (IFIPTM '13), AICT 401, pp. 1–16. Springer (2013)
99. Veeningen, M., Zannone, N., de Weger, B.: Formal Privacy Analysis of Communication Protocols for Identity Management. In: Proceedings of the 7th International Conference on Information Systems Security (ICISS '11), LNCS 7093, pp. 235–249. Springer (2011)
100. Vossaert, J., Lapon, J., De Decker, B., Naessens, V.: 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, pp. 155–170. Springer (2011)

A Trace Validity

In this appendix, we introduce "trace validity" as a way of verifying that all knowledge required for a trace has been modelled. Our

framework takes as input a trace, together with the initial knowledge of the actors. However, there are no guarantees that the trace and initial knowledge provided by the analyst are correctly specified. This is fundamental for the analysis, because the initial knowledge also determines whether an actor can link the information he has observed to information he already has. The concept of "trace validity" checks whether the initial knowledge and trace correspond to a valid scenario (i.e., a scenario in that can actually occur), and hence serves as a "sanity check" for the model.

To define trace validity, we need to model whether a context item has occurred in communication before. When an actor a initiates a protocol instance π in state $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$, 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 this freedom to instantiate context items because contents of the context items from $\mathsf{m}|^\pi$ he uses in his reply should corresponds to their contents in $\mathsf{m}|^\pi$ itself. In the former case, we call the context items *undetermined*; in the latter case, we call them *determined*:

Definition 13 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; or if p is a property $\psi_i(\mathsf{q})$ of some q occurring in m. Otherwise, p is *undetermined*.

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 $\mathsf{m}|^\pi$. We capture this by requiring that the actor can derive a message n that is equal to m, except that undetermined items are replaced by items of his choice. Intuitively, the actor having sufficient knowledge to send $\mathsf{m}|^\pi$ means that, when the message m is added to his knowledge base, he does not gain any new knowledge from this. For instance, if the actor can associate personal information from message $\mathsf{m}|^\pi$ to information in his knowledge base, then he should be able to make the same associations using the corresponding item in n. The restrictions on n in the definition below guarantee that this is indeed the case:

Definition 14 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 $\mathsf{n}\equiv\mathsf{m}$ such that $\mathcal{C}_a\vdash\mathsf{n}$, and the following conditions hold:

1. Whenever $\mathsf{m}@z$ is determined, then $\mathsf{m}@z=\mathsf{n}@z$;
2. Whenever $\mathsf{m}@z_1=\mathsf{m}@z_2$, then $\mathsf{n}@z_1=\mathsf{n}@z_2$;
3. If $\mathsf{m}@z=d|_k^\kappa$ $(k\neq\cdot)$ and some $e|_k^\eta\in\mathsf{I}^c\cup\mathsf{D}^c$ is determined, then $\mathsf{n}@z\leftrightarrow_a e|_k^\eta$;
4. If $\mathsf{m}@z_1=d|_k^\kappa$, $\mathsf{m}@z_2=d'|_k^\kappa$ $(k\neq\cdot)$, and no $e|_k^\eta\in\mathsf{I}^c\cup\mathsf{D}^c$ is determined, then $\mathsf{n}@z_1\leftrightarrow_a\mathsf{n}@z_2$.

Condition 1 states that the actor cannot replace determined items; condition 2 states that he should replace items consistently. Conditions 3 and 4 make sure that actors cannot learn new associations by using n as m: condition 3 applies to contexts already used in previous communication, and condition 4 applies to previously unused contexts. For determined messages, determinability and detectability coincide.

The following example demonstrates determinability:

Example 11 Consider the state $\{\mathcal{C}_x^0\}_{x\in\mathcal{A}}$ from Example 10. The client's message $\mathsf{m}=E'_{shkey|_\cdot}(id|_{su})|^\pi$ is determinable by *cli* in this state. Namely, take $\mathsf{n}=E'_{skey|_\cdot}(id|_4^{ab})$. Then $\mathsf{m}\equiv\mathsf{n}$, and this message trivially satisfies conditions 1–4 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 is

$$\mathcal{C}_{srv}^1=\mathcal{C}_{srv}^0\cup\{ip|_{cli}^\cdot,ip|_{srv}^\cdot,E'_{shkey|_\cdot}(id|_{su})|^\pi\},$$

and the server's reply is

$$\mathsf{m}=E'_{shkey|_\cdot}(\{age|_{su},n|_\cdot^\cdot,S_{k-|_{srv}}(\{age|_{su},n|_\cdot^\cdot\})\})|^\pi.$$

Indeed, one can verify that

$$\mathsf{n}=E'_{shkey|_\cdot^\pi}(\{col1|_1^{db},n|_\cdot^\cdot,S_{k-|_{srv}^\pi}(\{col1|_1^{db},n|_\cdot^\cdot\})\})$$

satisfies the conditions from the above definition. Namely, no determined items from m have been replaced in n (condition 1); both occurrences of $age|_{su}^\pi$ have been replaced by the same item, and similarly for $n|_\cdot^\pi$ (condition 2); and $col1|_1^{db}\leftrightarrow_{srv}id|_{su}^\pi$, i.e., the message contains only associations known by *srv* (condition 3). Condition 4 holds trivially because there are no two context items satisfying the given condition. □

Trace validity is defined step-by-step from the validity of its message transmissions. A message transmission consists of identifiers a,b of the communication parties and communicated message m. For validity, we require determinability both of the message, and of the communication identifiers. This way, we check that both the knowledge required to send the message, and the knowledge of where to send the message to, have been modelled. Formally, for a basic message transmission $\mathsf{a}\to\mathsf{b}:\mathsf{m}$, this means determinability by the sender of the context message $\{\mathsf{m},\mathsf{a},\mathsf{b}\}$. For the other two types of the form $\mathsf{a}\mapsto\mathsf{b}:\mathsf{m}$ modelling cryptographic protocols, both actors contribute information: the initiator of the protocol should determine the sender and receiver addresses a, b, and both parties contribute parts of m:

Definition 15 Let $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ be a state, and t a message transmission. Let $\mathsf{t}=\mathsf{a}\to\mathsf{b}:\mathsf{m}$ or $\mathsf{a}\mapsto\mathsf{b}:\mathsf{m}$, and let $a,b\in\mathcal{A}$ be the actors such that $a\leftrightarrow\sigma(\mathsf{a})$, $b\leftrightarrow\sigma(\mathsf{b})$. We say that t is *valid* in $\{\mathcal{C}_x\}_{x\in\mathcal{A}}$ if the messages indicated in Table 7 are determinable by a and b, respectively. 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 message transmission t_i is valid in respective state $\{\mathcal{C}_x^{i-1}\}_{x\in\mathcal{A}}$.

For ZK proofs, the prover needs to know the private information for the proof, and both parties contribute randomness. Note that to participate in the protocol, the verifier does not need to know the public information or the properties to be proven; however, he does need to know this information to be able to interpret the proof (i.e., to apply the testing rule). For credential issuing, the user needs to know her secret identifier m_1, randomness, and the issuer's public key; the issuer needs to know his private/public key pair, the attributes to be signed, and additional randomness.

The following example highlights validity of message transmissions and traces.

Example 12 Consider the trace given in Example 10. In Example 11, we showed determinability of the two messages transmitted in the trace; this argument can be easily extended to conclude determinability of the messages $\{\mathsf{a},\mathsf{b},\mathsf{m}\}$ from Definition 15, and hence validity of the two message transmissions. We conclude that the trace is valid. □

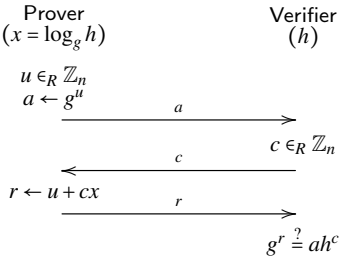
Trace validity is implemented in the tool supporting our framework. We briefly discuss the implementation. The main task in implementing trace validity is to check for determinability of a message m; that is, to find a derivable message n that is equivalent to m and satisfies properties (1) to (4) from Definition 14. Properties (1) and (2) place restrictions on the form of the message, which can be expressed in terms of free variables in a Prolog query to the deductive system. For properties (3) and (4) we check associability as in Section 3.3.

B Inference Rules for Zero-Knowledge Proofs and Credential Issuing

In this appendix, we show how our models of ZK proofs and the credential issuing protocol are derived.

t =	**Determinable by** a	**Determinable by** b
$\mathsf{a} \to \mathsf{b} : \mathsf{m}$	$\{\mathsf{a}, \mathsf{b}, \mathsf{m}\}$	$\varnothing$
$\mathsf{a} \mapsto \mathsf{b} : \mathrm{ZK}(\mathsf{m}_1; \mathsf{m}_2; \mathsf{m}_3; \{\mathsf{n}_a, \mathsf{n}_b\})$	$\{\mathsf{a}, \mathsf{b}, \mathsf{m}_1, \mathsf{n}_a\}$	n_b
$\mathsf{a} \mapsto \mathsf{b} : \mathrm{ICred}^{\mathsf{m}_1}_{\mathsf{m}_2}(\mathsf{m}_3; \{\mathsf{n}_j\}_{j=1}^{7})$	$\{\mathsf{a}, \mathsf{b}, \mathsf{pk}(\mathsf{m}_2), \mathsf{m}_1, \mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3, \mathsf{n}_7\}$	$\{\mathsf{pk}(\mathsf{m}_2), \mathsf{m}_2, \mathsf{m}_3, \mathsf{n}_4, \mathsf{n}_5, \mathsf{n}_6\}$

Table 7 Determinability requirements for the different types of message transmissions



(a) Schnorr proof of knowledge

$$\mathsf{p} \mapsto \mathsf{v} : \mathrm{ZK}(\mathsf{k}^-; \mathsf{pk}(\mathsf{k}^-); \varnothing; \{\mathsf{n}_p, \mathsf{n}_v\})$$

(where $\mathsf{k}^- = x$, $\mathsf{pk}(\mathsf{k}^-) = h$, $\mathsf{n}_p = u$, $\mathsf{n}_v = c$)

(b) Formal model of Schnorr proof

Fig. 19 Schnorr proof of knowledge and its formal model

B.1 Zero-Knowledge Proofs

ZK proofs allow a prover to prove to a verifier that he knows some secret information satisfying certain properties with respect to some public information, without revealing any information about the secret. For instance, consider a large group of prime order n generated by a group element g. Note that given value h, it is infeasible to determine the discrete logarithm $x = \log_g h$; this property can be exploited to build a public key cryptosystem in which values of h are public keys, and the corresponding values of x are private keys. A *prover* who knows x as well as n, g, and h can engage in a ZK proof protocol with a *verifier* who just knows n, g, and h; when the protocol has finished successfully, the verifier is convinced that the prover knows the value of x, without learning anything about its value.

The general definition of ZK proofs leaves open different kinds of implementations; we model a particular kind of ZK proof called Σ-protocols [41]. Σ-protocols are *three-move* protocols in which the prover first sends a *commitment*; the verifier responds with a randomly generated *challenge*; and finally the prover sends a *response*. The ZK proofs used in the systems analysed [8,24,25,51] are of this kind.

An example Σ-protocol is the protocol proposed by Schnorr to prove knowledge of $x = \log_g h$ in the setting given above (Figure 19(a)). The prover computes a random u and sends a commitment g^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 [85].

We formally model ZK proofs at a high level using the primitive $\mathrm{ZK}(\mathsf{m}_1; \mathsf{m}_2; \mathsf{m}_3; \mathsf{n})$. The secret information m_1 and public information m_2 are described in terms of messages; the ZK proof proves that the public information has a certain message structure with respect to the secret information. In addition, the proof can show that context data items d occurring in m_1 satisfy properties $\psi_k(\mathsf{d})$, listed in m_3. Finally, n represents randomness; in Σ-protocols, $\mathsf{n} = \{\mathsf{n}_p, \mathsf{n}_v\}$, representing the provers' randomness n_p for the commitment and the verifier's randomness n_v for the challenge. For instance, $\mathrm{ZK}(\mathsf{k}^-; \mathsf{pk}(\mathsf{k}^-); \varnothing; \{\mathsf{n}_p, \mathsf{n}_v\})$ is a proof of knowledge of the private key k^- corresponding to public key $\mathsf{pk}(\mathsf{k}^-)$ with no properties and contributed randomness $\mathsf{n}_p, \mathsf{n}_v$.

$$\frac{\mathcal{C}_a \vdash \mathrm{ZK}(\mathsf{m}_1; \mathsf{m}_2; \mathsf{m}_3; \{\mathsf{n}_p, \mathsf{n}_v\})}{\mathcal{C}_a \vdash \mathsf{m}_3} (\vdash\mathbf{EZ_1'})$$

$$\frac{\mathcal{C}_a \vdash \mathrm{ZK}(\mathsf{m}_1; \mathsf{m}_2; \mathsf{m}_3; \{\mathsf{n}_p, \mathsf{n}_v\})}{\mathcal{C}_a \vdash \mathsf{n}_v} (\vdash\mathbf{EZ_2'})$$

$$\frac{\mathcal{C}_a \vdash \mathrm{ZK}(\mathsf{m}_1; \mathsf{m}_2; \mathsf{m}_3; \{\mathsf{n}_p, \mathsf{n}_v\})}{\mathcal{C}_a \vdash \mathsf{m}_2} (\vdash\mathbf{EZ_3'})$$

$$\frac{\mathcal{C}_a \vdash \mathrm{ZK}(\mathsf{m}_1; \mathsf{m}_2; \mathsf{m}_3; \{\mathsf{n}_p, \mathsf{n}_v\}) \quad \mathcal{C}_a \vdash \mathsf{n}_p}{\mathcal{C}_a \vdash \mathsf{m}_1} (\vdash\mathbf{EZ_4'})$$

$$\frac{\mathcal{C}_a \vdash \mathrm{ZK}(\mathsf{m}_1; \mathsf{m}_2; \mathsf{m}_3; \{\mathsf{n}_p, \mathsf{n}_v\}) \quad \mathcal{C}_a \vdash \mathsf{m}_1}{\mathcal{C}_a \vdash \mathsf{n}_p} (\vdash\mathbf{EZ_5'})$$

$$\frac{\mathcal{C}_a \vdash \mathrm{ZK}(\mathsf{m}_1; \mathsf{m}_2; \mathsf{m}_3; \{\mathsf{n}_p, \mathsf{n}_v\}) \quad \mathcal{C}_a \vdash \mathsf{n}_p' \quad (\mathsf{n}_p' \doteq \mathsf{n}_p)}{\mathcal{C}_a \vdash \mathsf{n}_p} (\vdash\mathbf{TZ_1'})$$

$$\frac{\mathcal{C}_a \vdash \{\mathsf{m}_1, \ldots, \mathsf{m}_j, \mathsf{n}_1, \ldots, \mathsf{n}_k, \mathsf{p}_1, \ldots, \mathsf{p}_l, \mathsf{n}_p, \mathsf{n}_v\}}{\mathcal{C}_a \vdash \mathrm{ZK}(\{\mathsf{m}_1, \ldots, \mathsf{m}_j\}; \{\mathsf{n}_1, \ldots, \mathsf{n}_k\}; \{\mathsf{p}_1, \ldots \mathsf{p}_l\}; \{\mathsf{n}_p; \mathsf{n}_v\})} (\vdash\mathbf{CZ'})$$

Fig. 20 Complete set of inference and rules for ZK ($\mathcal{C}_a$ a set of context messages; m_*, n_* context messages; p_i properties of m_k, i.e., every $\mathsf{p}_i = \psi_j(\mathsf{m}_k) \in \mathsf{D}^c$ for some j, k))

From this high-level description in terms of structure of messages, the low-level description follows implicitly. For instance, in a setting where public/private key pairs are of the form $(h, x = \log_g h)$, the proof $\mathrm{ZK}(\mathsf{k}^-; \mathsf{pk}(\mathsf{k}^-); \varnothing; \{\mathsf{n}_p, \mathsf{n}_v\})$ corresponds to a proof of knowledge of the discrete logarithm $x = \log_g h$ of h like the Schnorr protocol. Figure 19 shows the Schnorr protocol and its formal model in this setting.

In Figure 20, we present a set of inference rules for the ZK primitive. We first explain them, and then argue that for privacy purposes and under certain assumptions, it suffices to consider the smaller set of rules presented in Figure 4. We first discuss what messages can be derived from a ZK transcript $\mathrm{ZK}(\mathsf{m}_1; \mathsf{m}_2; \mathsf{m}_3; \{\mathsf{n}_p, \mathsf{n}_v\})$ using elimination and testing rules. The property proven by a ZK proof determines the format of the messages in the ZK proof protocol. Hence, we allow any actor to derive the properties m_3 from the transcript (**$\vdash$EZ$_1$'**). (Because different properties may have identically-looking ZK proof protocols, this is an over-approximation of knowledge.) The verifier randomness n_v is transmitted as challenge, and so can be derived from the transcript (**$\vdash$EZ$_2$'**). Because both parties are assumed to know m_2 before the start of a ZK proof, it does not need to follow from the transcript. However, depending on the protocol, it may be possible to derive m_2. E.g., in the Schnorr example, $h = (g^r a^{-1})^{-c}$. Hence, as a possible over-approximation, we allow any observer to derive the public information m_2 (**$\vdash$EZ$_3$'**).

The fact that the protocol is zero-knowledge means that a verifier (who knows m_2, m_3 and n_v) should not be able to learn anything about m_1. In fact, if there are several possible secrets m_1 corresponding to public information m_2, then the probability distribution for protocol transcripts is required to be independent from m_1. Thus, it is impossible to test m_1 from the transcript. (Of course, if m_2 determines m_1, e.g., if they are a public/private key pair, then m_1 can be derived using m_2, but this is not due to the ZK proof.) Because the verifier, who knows all components of the ZK proof except m_1 and n_p, cannot deduce anything about the secret m_1, any inference rule to derive it needs to have n_p as

$$\frac{\mathcal{C}_a \vdash \mathsf{pk}(\mathsf{k}^-) \quad \mathcal{C}_a \vdash \mathsf{m}_1 \quad \mathcal{C}_a \vdash \mathsf{n}_a}{\mathcal{C}_a \vdash S^0_{\mathsf{k}^-}(\mathsf{m}_1, \mathsf{n}_a)} (\vdash\mathbf{CS}^0)$$

$$\frac{\mathcal{C}_a \vdash S^0_{\mathsf{k}^-}(\mathsf{m}_1, \mathsf{n}_a) \quad \mathcal{C}_a \vdash \{\mathsf{k}^-, \mathsf{m}_2, \mathsf{n}_b\}}{\mathcal{C}_a \vdash S_{\mathsf{k}^-}(\mathsf{m}_1, \mathsf{m}_2, \mathsf{n}_a, \mathsf{n}_b)} (\vdash\mathbf{CS}^{0\prime})$$

Fig. 21 Inference rules for signature scheme with signatures on committed values ($\mathcal{C}_a$ a set of context messages; k^-, m_*, n_* context messages)

a prerequisite. By a similar line of reasoning, if m_1 can be derived from n_p, then an inference rule for n_p needs m_1, or it needs to be a testing rule. In fact, in the Schnorr proof, in Σ-protocols all these inferences can be made: m_1 can be derived directly from n_p (**$\vdash$EZ$_4$'**) and vice versa (**$\vdash$EZ$_5$'**), and n_p can be tested (**$\vdash$TZ$_2$'**).

To generate a transcript $\mathrm{ZK}(\mathsf{m}_1;\mathsf{m}_2;\mathsf{m}_3;\{\mathsf{n}_p,\mathsf{n}_v\})$ of a Σ-protocol, an actor needs n_p for the commitment; n_v for the challenge; and both pieces of randomness and the private information for the response n_p (**$\vdash$CZ'**). (Technically, the public information is not needed.) Similarly, for determinability of the message transmission $\mathsf{a} \mapsto \mathsf{b} : \mathrm{ZK}(\mathsf{m}_1;\mathsf{m}_2;\mathsf{m}_3;\{\mathsf{n}_p,\mathsf{n}_v\})$, the prover needs $\{\mathsf{m}_1,\mathsf{n}_p\}$ in addition to the communication addresses $\{\mathsf{a},\mathsf{b}\}$; the verifier needs n_v.

There are two aspects the above model does not take into account. First, from two ZK proofs using the same prover randomness, the secret can be derived: in case of the Schnorr proof, 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*. However, if the prover always honestly generates his randomness, then this is very unlikely and we can safely ignore it. Second, an actor can also "simulate" a ZK proof transcript 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 ZK proof, but because the randomness in the commitment is unknown, it cannot be used to derive a secret corresponding to the public information. Such simulations are very unlikely to correspond to ZK proofs that really took place, so they are not relevant for knowledge analysis.

To express privacy requirements, the knowledge of randomness is not directly relevant. In addition, assuming that the randomness of the ZK proof is freshly generated and not reused elsewhere, it is clear that it cannot help to derive information indirectly: (**$\vdash$EZ$_4$'**) is the only rule to derive personal information (namely, m_1) using randomness, and it has knowledge of n_p as prerequisite, which can only be derived when m_1 is already known. Ignoring rules (**$\vdash$EZ$_2$'**), (**$\vdash$EZ$_5$'**), we obtain the inference rules given in Figure 4, and determinability requirements in Table 7.

B.2 Anonymous Credentials and Issuing

In an anonymous credential system, credentials $\mathrm{cred}^{M_1}_{\mathsf{k}^-}(M_2;M_3)$ assert the link between a user's identifier M_1 and her attributes M_2 using secret key k^-, and such credentials are issued and shown anonymously [24]. Anonymous issuing means the issuer of the credential does not learn the user's identifier M_1 (in particular, this means he cannot issue credentials containing the identifier without the user's involvement). We model the issuing protocol by the $\mathrm{ICred}^{M_1}_{\mathsf{k}^-}(M_2;M_3')$ primitive. The randomness M_3' used in the issuing protocol determines the randomness M_3 in the credential. Anonymous showing means that it is possible to perform ZK proofs of ownership of a credential proving certain properties. This is captured by our ZK primitive.

We model anonymous credential systems constructed from signature schemes [24,25] as used in the Identity Mixer system [8]. In general, this construction is possible if the signature scheme allows for issuing of signatures on committed values (Figure 21). That is, a commitment $S^0_{\mathsf{k}^-}(\mathsf{m}_1,\mathsf{n}_a)$ to message m_1 using randomness n_a is constructed using public key $\mathsf{pk}(\mathsf{k}^-)$ (**$\vdash$CS^0**); this commitment is turned into signature $S_{\mathsf{k}^-}(\mathsf{m}_1,\mathsf{m}_2,\mathsf{n}_a,\mathsf{n}_b)$ using private key k^-, message m_2 and randomness n_b, (**$\vdash$CS$^{0\prime}$**). Based on such a scheme, an anonymous credential $\mathrm{cred}^{\mathsf{m}_1}_{\mathsf{k}^-}(\mathsf{m}_2;\{\mathsf{n}_a,\mathsf{n}_b\})$ is simply a randomised signature (containing secret identifier m_1 and attributes m_2) along with its used randomness. In the Identity Mixer system, two such signature schemes can be used: SRSA-CL signatures [24] and BM-CL signatures [25]. There are slight technical differences between the two; we discuss SRSA-CL signatures and briefly outline the differences later.

$$\begin{aligned}
&\mathsf{a} \to \mathsf{b} : S^0_{\mathsf{k}^-}(\mathsf{m}_1,\mathsf{n}_2);\\
&\mathsf{a} \mapsto \mathsf{b} : \mathrm{ZK}(\mathsf{m}_1,\mathsf{n}_1,\mathsf{n}_2;\mathsf{pk}(\mathsf{k}^-),\mathcal{H}(\mathsf{m}_1,\mathsf{n}_1),S^0_{\mathsf{k}^-}(\mathsf{m}_1,\mathsf{n}_2);\varnothing;\{\mathsf{n}_3,\mathsf{n}_4\});\\
&\mathsf{b} \to \mathsf{a} : \{S_{\mathsf{k}^-}(\mathsf{m}_1,\mathsf{m}_2,\mathsf{n}_2,\mathsf{n}_5),\mathsf{n}_5\};\\
&\mathsf{b} \mapsto \mathsf{a} : \mathrm{ZK}(\mathsf{k}^-;\mathsf{pk}(\mathsf{k}^-),S^0_{\mathsf{k}^-}(\mathsf{m}_1,\mathsf{n}_2),\mathsf{m}_2,\mathsf{n}_5,\\
&\qquad S_{\mathsf{k}^-}(\mathsf{m}_1,\mathsf{m}_2,\mathsf{n}_2,\mathsf{n}_5);\varnothing;\{\mathsf{n}_6,\mathsf{n}_7\})
\end{aligned}$$

Credential obtained: $\{S_{\mathsf{k}^-}(\mathsf{m}_1,\mathsf{m}_2,\mathsf{n}_2,\mathsf{n}_5),\mathsf{n}_2,\mathsf{n}_5\}$

(a) Issuing protocol for anonymous credentials

$$\mathsf{a} \mapsto \mathsf{b} : \mathrm{ICred}^{\mathsf{m}_1}_{\mathsf{k}^-}(\mathsf{m}_2;\{\mathsf{n}_i\}_{i=1}^7)$$

Credential obtained: $\mathrm{cred}^{\mathsf{m}_1}_{\mathsf{k}^-}(\mathsf{m}_2;\{\mathsf{n}_2,\mathsf{n}_5\})$

(b) Formal model of anonymous credential issuing protocol

Fig. 22 Anonymous credentials from signature scheme with signatures on committed values

The anonymous credential issuing protocol can be modelled as a trace in terms of the signature scheme (Figure 22(a)). It involves a user a and an issuer b. As before, a is assumed to have sent a commitment $\mathcal{H}(\mathsf{m}_1,\mathsf{n}_1)$ to her secret identifier to b prior to initiating the protocol. (Unlike the commitment $S^0_{\mathsf{k}^-}(\mathsf{m}_1,\mathsf{n}_2)$ for the signature, $\mathcal{H}(\mathsf{m}_1,\mathsf{n}_1)$ does not depend on k^- and can thus be shared with other issuing or showing protocols for credentials having a different key.) In the first two messages, actor a provides her commitment for the signature, and then proves that it is formed correctly; that is, it indeed contains the identifier corresponding to the one in $\mathcal{H}(\mathsf{m}_1,\mathsf{n}_1)$. Actor b uses the commitment to construct a signature on $\{\mathsf{m}_1,\mathsf{m}_2,\mathsf{n}_2,\mathsf{n}_5\}$, and sends the signature along with his randomness to a. At this point, a knows the signature and the two pieces of randomness used in it: these three components together form the anonymous credential, as shown in the figure. (Note that b does not know n_2, so he does not have the complete credential.) In the last step, the signer b proves that $S_{\mathsf{k}^-}(\mathsf{m}_1,\mathsf{m}_2,\mathsf{n}_2,\mathsf{n}_5)$ is valid; when using the SRSA-CL signature scheme, this step is technically needed to ensure the security of the signature [8]. Figure 22(b) displays our high-level model of the issuing protocol and the credential obtained from it.

The high-level inference rules (Figure 4) and determinability relation (Table 7) for cred and ICred follow from the lower-level model in Figure 22(a). The credential's signature can be verified using messages $\{\mathsf{pk}(\mathsf{k}^-),\mathsf{m}_1,\mathsf{m}_2\}$, and a credential can be constructed from its components (**$\vdash$CR**). Although randomness can be inferred from the credential, we do not model these inferences in the high-level model because they are not relevant for knowledge of personal information.

From the issuing protocol, the user can infer the credential using the randomness from the credential (**$\vdash$EI$_1$**). We check the messages of the trace for further possible inferences. For the two ZK proofs, (**$\vdash$EZ$_1$**) does not apply because there are no proofs of properties. The (**$\vdash$EZ$_2$**) rule can be applied to both ZK proofs occurring in the issuing protocol; this translates to rules (**$\vdash$EI$_2$**) and (**$\vdash$EI$_3$**). We also consider the derivation of the nonces n_1, n_2 (**$\vdash$EI$_2$**): n_1 is generated outside of the issuing protocol, so its derivation may be of interest; n_2 is a pre-

requisite for ($\vdash\mathbf{EZ}_2$). Rule ($\vdash\mathbf{EZ}_3$) gives ($\vdash\mathbf{EI}_4$). We do not add a rule to derive $S^0_{\mathsf{k}^-}(\mathsf{m}_1,\mathsf{n}_2)$ from the transcript because its knowledge is not relevant from a privacy point of view. Also, this message does not allow the derivation of any information that was not already derivable from the zero-knowledge proofs. However, it does give testing rule ($\vdash\mathbf{TI}_2$). Testing rule rules ($\vdash\mathbf{TI}_1$) and ($\vdash\mathbf{TI}_3$) follow from the first message transmission. The other testing rules ($\vdash\mathbf{TI}_4$), ($\vdash\mathbf{TI}_5$) follow from the corresponding testing rule ($\vdash\mathbf{TZ}_1$) for zero-knowledge proofs.

Finally, consider $\mathrm{ICred}^{\mathsf{m}_1}_{\mathsf{k}^-}(\mathsf{m}_2;\{\mathsf{n}_i\}_{i=1}^7)$'s determinability requirements. Assuming fresh nonces, determinability of $\{\mathsf{a},\mathsf{b},\mathsf{pk}(\mathsf{k}^-),\mathsf{m}_1,\mathsf{n}_2\}$ by a is required for the first message transmission. 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 of $\{\mathsf{k}^-,\mathsf{m}_2,\mathsf{n}_5\}$ by b. The last ZK proof additionally means determinability of $\{\mathsf{pk}(\mathsf{k}^-),\mathsf{n}_6\}$ by b, and n_7 by a. We get the determinability requirements given in Table 7. Note that technically, a does not need m_2 to run the protocol, and b does not need $\mathcal{H}(\mathsf{m}_1,\mathsf{n}_1)$; however, in practice, they will check whether the data supplied matches their expectations using the checks expressed by the testing rules.

We mention two modelling details regarding the use of SRSA-CL signatures for anonymous credentials. First, the last ZK proof in the issuing trace is technically not a proof of knowledge of the private key, but of the RSA inverse of part of the issuer's randomness. However, in terms of knowledge this proof is equivalent because the private key can be determined from the RSA inverse and vice versa [18]. Second, due to the structure of the signature, different choices for n_a and n_b can lead to content equivalent signatures. However, assuming n_a and n_b are chosen at random, this happens with negligible probability.

Finally, an alternative signature scheme supporting signatures on committed values is the BM-CL scheme [25]. There are two technical differences with the SRSA-CL-based system presented above. First, BM-CL signatures have the additional property that they allow "blinding": a user can turn a valid credential $\mathrm{cred}^{\mathsf{m}_1}_{\mathsf{k}^-}(\mathsf{m}_2;\{\mathsf{n}_a,\mathsf{n}_b\})$ into a different credential $\mathrm{cred}^{\mathsf{m}_1}_{\mathsf{k}^-}(\mathsf{m}_2;\{\mathsf{n}'_a,\mathsf{n}_b\})$ (however, she is not able to change randomness n_b). Second, the final ZK proof in the issuing protocol of Figure 22 is not necessary for a BM-CL-based scheme. We chose the SRSA-CL-based signature scheme because the high-level model is simpler; however, in terms of privacy the choice of signature scheme does not matter.