Formal Privacy Analysis of Communication Protocols for Identity Management*

Meilof Veeningen, Benne de Weger, and Nicola Zannone

Eindhoven University of Technology, The Netherlands
{m.veeningen,b.m.m.d.weger,n.zannone}@tue.nl

Abstract. Over the years, formal methods have been developed for the analysis of security and privacy aspects of communication in IT systems. However, existing methods are insufficient to deal with privacy, especially in identity management (IdM), as they fail to take into account whether personal information can be linked to its data subject. In this paper, we propose a general formal method to analyze privacy of communication protocols for IdM. To express privacy, we represent knowledge of personal information in a three-layer model. We show how to deduce knowledge from observed messages and how to verify a range of privacy properties. We validate the approach by applying it to an IdM case study.

1 Introduction

With the growth of social networking, e-business, e-Government, and ubiquitous computing, more and more personal information is being handled over the Internet. This has increased the need to design IT systems that preserve user privacy. Not only users may demand that the IT systems they interact with preserve their privacy, but also privacy regulations (such as the EU Data Protection Directive) impose stringent requirements on the collection, processing, and disclosure of personal information.

Identity management (IdM) [1,2,3] is an emerging technology for handling personal data in distributed systems. In such a system, a service provider (SP) retrieves user credentials from (possibly multiple) identity providers (IdPs) for the authentication of users, leading to an exchange of information which may also involve the user or third parties. This information exchange impacts the user's privacy: the design of the protocols used by parties to communicate determines how much personal information is learned by the various parties, and to what extent they can link these different pieces of information. This makes it important to compare these protocols in a precise way.

Over the years, formal methods have arisen as an important tool to analyze security of communication in IT systems [4,5,6,7]. The idea is to express communication protocols in a suitable formal model, and then verify whether such a model satisfies properties such as authentication properties [5] and secrecy properties [8]. Secrecy properties, in particular, can be used to express one aspect of privacy; namely, whether a certain piece of information is known by some party in a protocol. However, they can not be used to express another fundamental aspect of privacy; namely, to what extent a piece of personal information is linkable to the corresponding data subject (who, in general, may

* This work is partially funded by the Dutch Sentinel Mobile IDM project (#10522).

S. Jajodia and C. Mazumdar (Eds.): ICISS 2011, LNCS 7093, pp. 235–249, 2011.

not even participate directly in the protocol). Recently, formal methods have been extended to address privacy issues. However, in some cases the properties defined and verified are specific to their respective settings such as e-voting [9]. In other cases [10,11], the focus is on linking messages rather than interpreting them as personal information about a data subject as needed for the analysis of IdM systems.

In our previous work [12], we captured privacy in a general formal model for knowledge of personal information. This model expresses to which entity different pieces of information belong, and what knowledge actors have about these items and their relations. Based on this representation, we formally defined and compared identity-related properties, e.g., anonymity, pseudonymity and identifiability. However, the model cannot capture how this knowledge follows from communication as it does not allow interpretation of personal information in terms of how it was learned or what it contains.

In this paper, we combine existing formal methods and our previous work [12] by presenting a framework for analyzing which identity-related properties are satisfied by a system, given the information observed from communication protocols. This provides the machinery to compare the privacy of communication protocols in various IdM architectures in a precise way. The contributions of this paper are as follows:

- We define a *three-layer model* of (personal) information, which captures that (i) personal information in different contexts may satisfy different privacy properties; and (ii) different pieces of information may have the same contents.
- We take a representative set of cryptographic primitives and show how existing *deductive methods* for these primitives can (with some modifications) operate within our three-layer model.
- We show how to represent an actor's knowledge of personal information in terms of which personal information he can *detect*, and which personal information he can *associate* to a data subject.
- We verify, by checking these associations, which *identity-related properties*, as defined in our previous work [12], hold in a particular situation.

We demonstrate our approach by applying it to the attribute aggregation infrastructure proposed in the TAS^3 project. This infrastructure aims to satisfy a number of privacy properties: we check whether these privacy properties indeed hold, report on some problems we found, and provide some recommendations to improve the system.

The structure of the paper is as follows. We first introduce the three-layer model of personal information (§2). We use it to analyze what personal information an actor can deduce (§3) and associate (§4) from observed messages, and show how identity-related properties are defined and verified in terms of the model (§5). We apply our approach to the TAS^3 attribute aggregation infrastructure (§6), and present conclusions and directions for future work (§7).

2 A Three-Layer Model of Personal Information

In this section, we introduce a model for the representation and analysis of personal information that may be known by various actors within a system. The model can be seen as a refinement of the model proposed in our previous work [12], and is used to define actor knowledge (§4) and privacy properties (§5).

2.1 Personal Information

A piece of personal information in the digital world is a *specific* string that has a *specific* meaning as personal information about a *specific* person. 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. The set $\mathcal{E}$ of *entities* models the real-world persons whom the considered information is about.

The link between the information and its subject is captured by the *related* relation, denoted $\leftrightarrow$. This is an equivalence relation on entities, identifiers and data items, such that $o_1 \leftrightarrow o_2$ means that o_1 and o_2 are information about the same person. In particular, any identifier or data item is related to exactly one entity. Elements of the set $\mathcal{O} := \mathcal{E} \cup \mathcal{I} \cup \mathcal{D}$ are called *items of interest*.

These concepts, however, are insufficient to fully characterize the system dynamics. When interacting with a system, an actor may learn the same personal information several times without realizing that it is the same information. For example, consider two profiles of the same user that both contain "age=18", and suppose an actor does not know that the profiles are related. Then, from a privacy point of view (e.g., to check linkability between information in profiles) it is important to differentiate in the actor's knowledge between the age in the one profile and the age in the other profile.

In addition, an actor may be able to deduce information from the fact that different pieces of information have the same string contents. For example, if an actor encounters the same hash string in different contexts, and he knows the contents used in the first context, then he knows that these contents were also used in the second context.

2.2 Three-Layer Model

Because of the need to distinguish different instances of the same piece of information, but also to reason about message contents, we introduce a three-layer representation of personal information. The representation consists of the *object layer*, *information layer*, and *contents layer*. In the information layer, as described above, the information itself is represented, e.g., "the age of actor c". In the object layer, information is described along with the context in which it has been observed, e.g., "the age of the data subject in instance 1 of protocol π". In the contents layer, information is described in terms of the strings actually transmitted in a protocol, e.g., "age=18".

In the object layer, we model the *context* in which an actor knows pieces of information. A context is a tuple (η, k), where η is a *domain* and k is a *profile* within that domain. The sets of domains and profiles depend on the application; we deliberately do not define these sets here but instead content ourselves with some examples. One example domain could be $\phi =$ Facebook, in which the context $(\phi, 132)$ represents the profile of a particular Facebook user. Another example domain is "instance 2 of protocol π". In that domain, every party involved in the protocol is characterized by a profile.

In such a context, pieces of information are represented by *variables*. This representation makes it possible to reason about such personal information without regarding the instantiation. *Data item variables* represent data items (set D), whereas *identifier variables* represent identifiers (set I); consider, e.g., a variable $\mathit{age} \in \mathsf{D}$ denoting the age in a profile. A *context data item* is a data item variable d in a context (η, k), and we denote it $d|_k^\eta \in \mathsf{D}^c$; the set I^c of *context identifiers* is defined similarly. Entities are not represented by variables; instead, an entity $e \in \mathcal{E}$ in a context (η, k) is denoted $e|_k^\eta$; the set of *context entities* is $\mathcal{E}^c$. The reason is that, because entities are not digital information, there cannot be multiple "instances" of an entity. Every context contains exactly one entity who is the data subject, i.e., all information in the context belongs to that entity. $\mathcal{O}^c := \mathcal{E}^c \cup \mathsf{I}^c \cup \mathsf{D}^c$ is the set of *context items of interest*.

Items in the contents layer can be seen as strings of arbitrary length in some alphabet, i.e., the set Σ^*. The exact form of the contents layer is not relevant for our purposes. Rather, it *is* relevant to determine whether two pieces of information have the same contents: this is expressed using the τ function, as described below.

2.3 Maps between Layers and Equivalence

The link between the object layer and the information layer is given by the *substitution* $\sigma : \mathcal{O}^c \to \mathcal{O}$. We write σ as a list of context item-information pairs and application of σ in postfix notation, e.g., $\sigma = \{d|_k^\eta \to age_c, d'|_k^\eta \to haircolor_c\}$ and then $d|_k^\eta\sigma = age_c$. σ satisfies the following four properties: 1. $\sigma(\mathsf{D}^c) \subset \mathcal{D}$; 2. $\sigma(\mathsf{I}^c) \subset \mathcal{I}$; 3. $e|_k^\eta\sigma = e$ for any entity e, context (η, k); 4. $x|_k^\eta\sigma \leftrightarrow y|_k^\eta\sigma$ for any context items $x|_k^\eta, y|_k^\eta\sigma$. Intuitively, σ maps: 1. context data items to data items; 2. context identifiers to identifiers; 3. context entities to entities; 4. context items from the same context to related items of interest.

The link between information and its contents is given by function τ. The domain of the function is $\mathcal{I} \cup \mathcal{D}$ (entities have no contents). Function τ is injective on $\mathcal{I}$: this formally expresses the uniqueness of identifiers within the system.

We introduce notation for two context items $x|_k^\eta$, $y|_l^\chi$ representing the same information or contents. If $x|_k^\eta\sigma = y|_l^\chi\sigma$, then we write $x|_k^\eta \equiv y|_l^\chi$ and we call $x|_k^\eta$ and $y|_l^\chi$ *equivalent*. If $\tau(x|_k^\eta\sigma) = \tau(y|_l^\chi\sigma)$, then we write $x|_k^\eta \doteq y|_l^\chi$ and we call them *content equivalent*. Clearly, equivalence implies content equivalence. Two identifiers are equivalent iff they are content equivalence because of the injectivity of τ on identifiers.

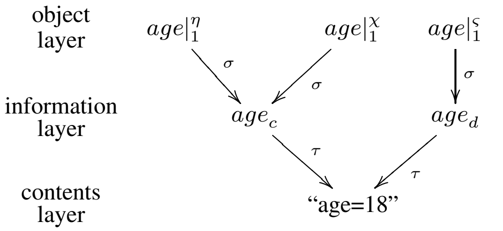
Example 1. Consider the three context messages $age|_1^\eta$, $age|_1^\chi$, and $age|_1^\varsigma$ in Fig. 1 where $age \in \mathsf{D}$. Let $\sigma = \{age|_1^\eta \to age_c, age|_1^\chi \to age_c, age|_1^\varsigma \to age_d\}$ with $\tau(age_c) = \tau(age_d) =$ "age=18". Then, $age|_1^\eta$ and $age|_1^\chi$ are equivalent; moreover, all three context messages given are content equivalent. □

3 Knowledge Analysis

In this section, we analyze how personal information can be derived from the messages that a user has observed. Deductive systems are often adopted for this purpose. We present a standard deductive system, and show how it can be adapted to the three-layer model. We also show that this adaptation does not impact its expressiveness.

3.1 Messages Analysis on the Information Layer

We present a formalism of messages and a deductive system similar to the ones usually adopted in protocol analysis [13]. Standard message analysis can be seen, in terms of our three-layer model, as operating on the information layer.



$$\begin{tikzcd}
\mathcal{O}^c = \mathcal{E}^c \cup \mathsf{I}^c \cup \mathsf{D}^c & \\
\mathsf{I}^c \cup \mathsf{D}^c \cup \mathcal{V}^c = \mathsf{P}^c \arrow[r, hook] \arrow[dd, "\sigma"] & \mathcal{L}^c \arrow[dd, "\sigma"] \\
\mathcal{O} = \mathcal{E} \cup \mathcal{I} \cup \mathcal{D} & \\
\mathcal{I} \cup \mathcal{D} \cup \mathcal{G} = \mathcal{P} \arrow[r, hook] \arrow[d, "\tau"] & \mathcal{L}^i \arrow[d, "\tau"] \\
\text{(actual string)} \arrow[r, hook] & \mathcal{L}^{cnt}
\end{tikzcd}$$

Fig. 1. Example of the three-layer model: three different context items with the information and contents they represent (left); the three-layer model of information (right)

Messages. The basic components of messages in communication protocols are *information items*. Apart from the sets $\mathcal{D}$ of data items and $\mathcal{I}$ of identifiers, we also consider a set $\mathcal{G}$ of non-personal information, such as shared keys and nonces. The set of information items is denoted $\mathcal{P} := \mathcal{D} \cup \mathcal{I} \cup \mathcal{G}$. Private and public keys are particular cases of identifiers. Private keys form a set $\mathcal{K}^- \subset \mathcal{I}$, public keys form a set $\mathcal{K}^+ \subset \mathcal{I}$, and, given a private key k^-, the corresponding public key is k^+ and vice versa.

Messages can be constructed from information items using cryptographic primitives. The set of *information messages*, denoted $\mathcal{L}^i$, is given by the following grammar:

$$M, N ::= p \mid E_{k^+}(M) \mid E'_N(M) \mid S_{k^-}(M) \mid \mathcal{H}(M) \mid \{M, N\} \tag{1}$$

where $p \in \mathcal{P}$, $k^+ \in \mathcal{K}^+$, $k^- \in \mathcal{K}^-$. This models, respectively: asymmetric encryption $E_{k^+}(M)$ of message M with public key k^+, symmetric encryption $E'_N(M)$ of message M with key N, signature $S_{k^-}(M)$ over message M with private key k^-, hash $\mathcal{H}(M)$ of message M, and (associative) concatenation $\{M, N\}$ of messages M and N.

We assume that these cryptographic primitives satisfy a number of properties. First, all primitives are deterministic; that is, given the same inputs, they always give the same output. Randomness in non-deterministic encryption or signing should be modeled explicitly as part of the plaintext. By signing we mean "clear-signing" [14]; that is, the message M can be recovered from $S_{k^-}(M)$ without knowledge of the corresponding public key k^+. (This can be achieved by appending the message to the "raw" signature.)

Finally, we assume *structural equivalence*. Extend τ from $\mathcal{P}$ to $\mathcal{L}^i$ by applying it to all information items in a message, e.g.: $\tau(E'_d(d')) = E'_{\tau(d)}(\tau(d'))$. The image $\tau(\mathcal{L}^i)$ is the language $\mathcal{L}^{cnt}$ generated by grammar (1) with contents instead of information items. Different elements of $\mathcal{L}^{cnt}$ could a priori be the same as strings, e.g. a collision in the hash function could cause $\mathcal{H}(\tau(x))$ and $\mathcal{H}(\tau(y))$ to be the same string even if $\tau(x) \neq \tau(y)$; or $E'_{\tau(x)}(\tau(y))$ could happen to be the same string as $\mathcal{H}(\tau(z))$. Structural equivalence is the assumption that this does not happen, i.e., the grammar $\mathcal{L}^{cnt}$ uniquely represents message contents. As a map from $\mathcal{L}^i$ to $\mathcal{L}^{cnt}$, τ satisfies two properties: 1. τ is injective on $\mathcal{I}$; 2. τ preserves the grammar structure of information messages.

Deductive System. A deductive system on $\mathcal{L}^i$ models which information messages $m \in \mathcal{L}^i$ an actor can *deduce* from the set $\mathcal{C}^i_a \subset \mathcal{L}^i$ of messages he knows (denoted $\mathcal{C}^i_a \vDash m$). Such a deductive system consists of a set of axioms and inference rules that mimic the idealized operation of the cryptographic primitives [13].

Axiom (⊨**0**) $$\frac{}{\mathcal{C}_a^i \models m}\ (m \in \mathcal{C}_a^i)\ (\models\mathbf{0})$$

Construction (⊨**C***) $$\frac{\mathcal{C}_a^i \models m \quad \mathcal{C}_a^i \models n}{\mathcal{C}_a^i \models \{m,n\}}\ (\models\mathbf{CC})$$

$$\frac{\mathcal{C}_a^i \models m}{\mathcal{C}_a^i \models \mathcal{H}(m)}\ (\models\mathbf{CH}) \qquad \frac{\mathcal{C}_a^i \models m \quad \mathcal{C}_a^i \models n}{\mathcal{C}_a^i \models E'_n(m)}\ (\models\mathbf{CE}) \qquad \frac{\mathcal{C}_a^i \models m \quad \mathcal{C}_a^i \models k^+}{\mathcal{C}_a^i \models E_{k^+}(m)}\ (\models\mathbf{CA})$$

$$\frac{\mathcal{C}_a^i \models m \quad \mathcal{C}_a^i \models k^-}{\mathcal{C}_a^i \models S_{k^-}(m)}\ (\models\mathbf{CS})$$

Elimination (⊨**E**) $$\frac{\mathcal{C}_a^i \models \{m,n\}}{\mathcal{C}_a^i \models m}\ (\models\mathbf{EC}) \qquad \frac{\mathcal{C}_a^i \models \{m,n\}}{\mathcal{C}_a^i \models n}\ (\models\mathbf{EC'})$$

$$\frac{\mathcal{C}_a^i \models E'_n(m) \quad \mathcal{C}_a^i \models n}{\mathcal{C}_a^i \models m}\ (\models\mathbf{EE}) \qquad \frac{\mathcal{C}_a^i \models E_{k^+}(m) \quad \mathcal{C}_a^i \models k^-}{\mathcal{C}_a^i \models m}\ (\models\mathbf{EA}) \qquad \frac{\mathcal{C}_a^i \models S_{k^-}(m)}{\mathcal{C}_a^i \models m}\ (\models\mathbf{ES})$$

Fig. 2. Deductive system on information ($\mathcal{C}_a^i \subset \mathcal{L}^i, m, n \in \mathcal{L}^i, k^+ \in \mathcal{K}^+, k^- \in \mathcal{K}^-$)

Fig. 2 shows a standard deductive system for information messages. The (⊨**0**) axiom expresses the deduction of any message in the set of known messages. The (⊨**C***) inference rules express the construction of concatenations, hashes, symmetric encryptions, asymmetric encryptions and signatures of deduced messages. The (⊨**E***) inference rules express decomposition of concatenations, decryption of symmetric and asymmetric encryptions whose key is known, and recovery of the plaintext from a signed message.

In the case of decryption, note that the deductive system does not express *how* the actor knows the decryption key, only *that* he knows it. Thus, an actor can try out any key to decrypt a message; if it happens to be the correct one, he obtains the plaintext. This means that the system over-estimates the knowledge of the actor in case he cannot actually tell by decrypting whether he used the right key or not, e.g. if the plaintext is something that is unknown, random, and unformatted such as a nonce.

Other properties of idealized cryptographic primitives are expressed by the absence of additional inference rules: e.g., one-wayness of hashes is accounted for by the absence of a rule to deduce m from $\mathcal{H}(m)$. In addition, note that there is no signature verification rule. This is because deductive systems focus on making deductions from known messages rather than checking message validity.

The *deduction* of a message using these rules is usually denoted in tree form. For example, we represent a deduction of age_c from $\mathcal{C}_a^i = \{E'_{key}(age_c), key\}$ as follows:

$$\frac{\dfrac{}{\mathcal{C}_a^i \models E'_{key}(age_c)}\ (\models\mathbf{0}) \qquad \dfrac{}{\mathcal{C}_a^i \models key}\ (\models\mathbf{0})}{\mathcal{C}_a^i \models age_c}\ (\models\mathbf{EE})$$

3.2 Message Analysis on the Object Layer

The deductive system above models the actor's knowledge on the information layer. However, for privacy analysis we need to distinguish between information from various contexts and reason about message contents. To achieve this, we adjust the deductive system to work on the object layer.

Messages. We define the set P of *context items* at the object layer analogously to the set $\mathcal{P}$ of information items. That is, $\mathsf{P} := \mathsf{D}^c \cup \mathsf{I}^c \cup \mathsf{V}^c$, with D^c and I^c the sets of context data items and identifiers. Similarly, V^c is the set of *context global items*, which can represent any information message, in particular items in $\mathcal{G}$. Context global items belong to a domain, but not to a profile; an example context global item is $shakey|^{\eta}_{\cdot}$.

The set $\mathcal{L}^c$ of *context messages* is generated by grammar (1), except that here p is any context item, and $k^+ \in \mathsf{K}^{+c} \subset \mathsf{I}^c$ and $k^- \in \mathsf{K}^{-c} \subset \mathsf{I}^c$ are context identifiers representing public and private keys, respectively. Notationally, contexts, domains, and profiles can be applied to messages, indicating application to all context items in the message, e.g., $E_{shakey|_{\cdot}}(age|_1)|^{\eta} := E_{shakey|^{\eta}_{\cdot}}(age|^{\eta}_1)$ and $\{id, age\}|^{\eta}_1 := \{id|^{\eta}_1, age|^{\eta}_1\}$.

The substitution σ extends from context items of interest to context messages in a natural way, e.g.: $\{\mathsf{m}_1, \mathsf{m}_2\}\sigma := \{\mathsf{m}_1\sigma, \mathsf{m}_2\sigma\}$. As a map from $\mathcal{L}^c$ to $\mathcal{L}^i$, σ satisfies properties 1–4 discussed in Section 2 as well as two additional properties: 5. $\sigma(\mathsf{K}^{+c}) \subset \mathcal{K}^+$, $\sigma(\mathsf{K}^{-c}) \subset \mathcal{K}^-$, and $key^+|^{\eta}_k\sigma = k^+$ iff $key^-|^{\eta}_k\sigma = k^-$ where k^- and k^+ are a private/public key pair; 6. σ preserves the grammar structure of context messages.

Sets $\mathcal{L}^c$, $\mathcal{L}^i$, and $\mathcal{L}^{cnt}$ and functions σ, τ form a three-layer model of messages that extends the personal information model (Fig. 1, right). Like context items, context messages m and n are *equivalent* iff $\mathsf{m}\sigma{=}\mathsf{n}\sigma$, and *content equivalent* iff $\tau(\mathsf{m}\sigma){=}\tau(\mathsf{n}\sigma)$.

Deductive System. To perform deduction on $\mathcal{L}^c$, we translate the inference rules on $\mathcal{L}^i$ to $\mathcal{L}^c$, but this is insufficient for two main reasons. First, although the object layer distinguishes between keys used in different contexts, an actor can re-use a key from one context in another. Second, an actor may infer additional information from the fact that different context messages have the same contents. We address the first problem with "key testing" rules, and the second with a "content analysis" rule.

The deductive system on the object layer (Fig. 3) models which context messages m an actor a can deduce from his known messages $\mathcal{C}_a \subset \mathcal{L}^c$ ($\mathcal{C}_a \vdash \mathsf{m}$). The rules **(⊢0)** to **(⊢EA)** are direct translations from the rules **(⊨0)** to **(⊨EA)** on the information layer. We now describe the additional object layer rules.

Key testing accounts for an actor knowing the key for decryption or signature verification of a message m, but not in the message's context. Note that in this case, e.g., decryption rule **(⊢EE)** can not be used directly. The key testing rules allow an actor, as in the deductive system on information, to try out on m any key he knows. If he uses a key with the correct contents, then he learns that it is the decryption **(⊢TA)**, **(⊢TE)** or signature verification **(⊢TS)** key. (Then, he can decrypt using **(⊢EE)** or **(⊢EA)**.) Note that in an implementation, to decide whether **(⊢T*)** can be applied, we only need to check the existence of a derivable content equivalent key, regardless of its context layer representation. For this, standard deduction techniques at the content layer suffice.

Example 2. Let $\mathcal{C}_a = \{E'_k(goods)|^{\pi}_{\cdot}, l|^{\rho}_{\cdot}\}$ be the set of messages known by an actor a, with $k|^{\pi}_{\cdot} \doteq l|^{\rho}_{\cdot}$. Then $\mathcal{C}_a \vdash goods|^{\pi}_{\cdot}$ can be deduced as follows:

$$\dfrac{\dfrac{}{\mathcal{C}_a \vdash E'_k(goods)|^{\pi}_{\cdot}}\,(\vdash\mathbf{0}) \qquad \dfrac{\dfrac{}{\mathcal{C}_a \vdash E'_k(goods)|^{\pi}_{\cdot}}\,(\vdash\mathbf{0}) \qquad \dfrac{}{\mathcal{C}_a \vdash l|^{\rho}_{\cdot}}\,(\vdash\mathbf{0})}{\mathcal{C}_a \vdash k|^{\pi}_{\cdot}}\,(\vdash\mathbf{TE})}{\mathcal{C}_a \vdash goods|^{\pi}_{\cdot}}\,(\vdash\mathbf{DE})$$

Axiom (⊢0)
$$\frac{}{\mathcal{C}_a \vdash \mathsf{m}}\ (\mathsf{m} \in \mathcal{C}_a)\ (\vdash\mathbf{0})$$

Construction (⊢C*)
$$\frac{\mathcal{C}_a \vdash \mathsf{m} \quad \mathcal{C}_a \vdash \mathsf{n}}{\mathcal{C}_a \vdash \{\mathsf{m},\mathsf{n}\}}\ (\vdash\mathbf{CC})$$

$$\frac{\mathcal{C}_a \vdash \mathsf{m}}{\mathcal{C}_a \vdash \mathcal{H}(\mathsf{m})}\ (\vdash\mathbf{CH}) \qquad \frac{\mathcal{C}_a \vdash \mathsf{m} \quad \mathcal{C}_a \vdash \mathsf{n}}{\mathcal{C}_a \vdash E'_{\mathsf{n}}(\mathsf{m})}\ (\vdash\mathbf{CE}) \qquad \frac{\mathcal{C}_a \vdash \mathsf{m} \quad \mathcal{C}_a \vdash \mathsf{k}^+}{\mathcal{C}_a \vdash E_{\mathsf{k}^+}(\mathsf{m})}\ (\vdash\mathbf{CA})$$

$$\frac{\mathcal{C}_a \vdash \mathsf{m} \quad \mathcal{C}_a \vdash \mathsf{k}^-}{\mathcal{C}_a \vdash S_{\mathsf{k}^-}(\mathsf{m})}\ (\vdash\mathbf{CS})$$

Elimination (⊢E*)
$$\frac{\mathcal{C}_a \vdash \{\mathsf{m},\mathsf{n}\}}{\mathcal{C}_a \vdash \mathsf{m}}\ (\vdash\mathbf{EC}) \qquad \frac{\mathcal{C}_a \vdash \{\mathsf{m},\mathsf{n}\}}{\mathcal{C}_a \vdash \mathsf{n}}\ (\vdash\mathbf{EC'})$$

$$\frac{\mathcal{C}_a \vdash E'_{\mathsf{n}}(\mathsf{m}) \quad \mathcal{C}_a \vdash \mathsf{n}}{\mathcal{C}_a \vdash \mathsf{m}}\ (\vdash\mathbf{EE}) \qquad \frac{\mathcal{C}_a \vdash S_{\mathsf{k}^-}(\mathsf{m})}{\mathcal{C}_a \vdash \mathsf{m}}\ (\vdash\mathbf{ES}) \qquad \frac{\mathcal{C}_a \vdash E_{\mathsf{k}^+}(\mathsf{m}) \quad \mathcal{C}_a \vdash \mathsf{k}^-}{\mathcal{C}_a \vdash \mathsf{m}}\ (\vdash\mathbf{EA})$$

Key testing (⊢T*)
$$\frac{\mathcal{C}_a \vdash E_{\mathsf{k}^+}(\mathsf{m}) \quad \mathcal{C}_a \vdash \mathsf{k'}^-}{\mathcal{C}_a \vdash \mathsf{k}^-}\ (\mathsf{k}^- \doteq \mathsf{k'}^-)\ (\vdash\mathbf{TA})$$

$$\frac{\mathcal{C}_a \vdash S_{\mathsf{k}^-}(\mathsf{m}) \quad \mathcal{C}_a \vdash \mathsf{k'}^+}{\mathcal{C}_a \vdash \mathsf{k}^+}\ (\mathsf{k}^+ \doteq \mathsf{k'}^+)\ (\vdash\mathbf{TS}) \qquad \frac{\mathcal{C}_a \vdash E'_{\mathsf{n}}(\mathsf{m}) \quad \mathcal{C}_a \vdash \mathsf{n}'}{\mathcal{C}_a \vdash \mathsf{n}}\ (\mathsf{n} \doteq \mathsf{n}')\ (\vdash\mathbf{TE})$$

Content analysis (⊢C)
$$\frac{\mathcal{C}_a \vdash \mathsf{m}_1 \quad \mathcal{C}_a \vdash \mathsf{m}_2 \quad \mathcal{C}_a \vdash \mathsf{n}_1}{\mathcal{C}_a \vdash \mathsf{n}_2}\ \begin{array}{l}((\mathsf{m}_1 \doteq \mathsf{m}_2) \Rightarrow (\mathsf{m}_3 \doteq \mathsf{m}_4);\\ \mathsf{n}_1 =_{\mathsf{m}_3 \sim \mathsf{m}_4} \mathsf{n}_2)\end{array}\ (\vdash\mathbf{C})$$

Fig. 3. Deductive system on context messages ($\mathcal{C}_a$ a set of context messages, m, m_i, n, n_i context messages; $\mathsf{k}^+/\mathsf{k}^-$ and $\mathsf{k'}^+/\mathsf{k'}^-$ public/private key pairs, $\Rightarrow$ as in Def. 2, $\mathsf{n}_1 =_{\mathsf{m}_3 \sim \mathsf{m}_4} \mathsf{n}_2$ means n_1 and n_2 are equal up to replacing m_3 by m_4 and vice versa)

The deduction models an actor testing whether $l|^{\rho}_{\cdot}$ is the decryption key for $E'_k(goods)|^{\pi}_{\cdot}$ (⊢**TE**). By learning it, the actor can decrypt the message (⊢**DE**). □

Content analysis lets an actor derive an unknown message from one context by concluding that it has the same contents as a known message from another. The statement of the rule relies on the syntactic structure of messages, which we first elaborate on.

The syntactic structure of messages describes the way they are constructed using cryptographic primitives. Primitives build up a message m given two (or, in the case of the hash, one) messages n and n': we define one to be the "left part" $\mathsf{n} = \mathsf{m}@l$ and the other to be the "right part" $\mathsf{n}' = \mathsf{m}@r$. Recursively, every submessage of m has a well-defined "position" in m:

Definition 1. *Let* m *be a context message and* $z \in \{l, r\}^*$. *Then,* $\mathsf{m}@z$, *the* submessage of m at z, *is defined as follows:* $\mathcal{H}(\mathsf{m})@l = \mathsf{m}$; $\{\mathsf{m}, \mathsf{n}\}@l = \mathsf{m}$; $\{\mathsf{m}, \mathsf{n}\}@r = \mathsf{n}$; $E'_{\mathsf{n}}(\mathsf{m})@l = \mathsf{n}$; $E'_{\mathsf{n}}(\mathsf{m})@r = \mathsf{m}$; $E_{\mathsf{k}^+}(\mathsf{m})@r = \mathsf{k}^+$; $E_{\mathsf{k}^+}(\mathsf{m})@r = \mathsf{m}$; $S_{\mathsf{k}^-}(\mathsf{m})@r = \mathsf{k}^-$; $S_{\mathsf{k}^-}(\mathsf{m})@r = \mathsf{m}$; $\mathsf{m}@z_1 ... z_n = ((\mathsf{m}@z_1)@...)@z_n$.

Note that for arbitrary context message m and $z \in \{l, r\}^*$, $\mathsf{m}@z$ may not be defined. For instance, $\mathcal{H}(x)@l$ is defined (and equal to x), but $\mathcal{H}(x)@r$ is not.

If two context messages m_1 and m_2 are content equivalent, then the properties of σ and τ imply content equivalence of their submessages. In particular, if $\mathsf{m}_1@z$ and $\mathsf{m}_2@z$ are both defined, then they are content equivalent. Also, if $\mathsf{m}_1@z = \mathsf{k}^+$ and $\mathsf{m}_2@z = \mathsf{k'}^+$, then not only $\mathsf{k}^+ \doteq \mathsf{k'}^+$ follows, but also $\mathsf{k}^- \doteq \mathsf{k'}^-$, and vice versa. The following notation expresses this intuition:

Definition 2. *Let* m_1, m_2, m_3, m_4 *be context messages. We write* $(\mathsf{m}_1 \doteq \mathsf{m}_2) \Rightarrow (\mathsf{m}_3 \doteq \mathsf{m}_4)$*, if* $\mathsf{m}_1 \doteq \mathsf{m}_2$ *and for some* $z \in \{l, r\}^*$*:*

- $\mathsf{m}_3 = \mathsf{m}_1@z$, $\mathsf{m}_4 = \mathsf{m}_2@z$*; or*
- $\mathsf{m}_1@z$, $\mathsf{m}_2@z$ *represent public keys of which* m_3, m_4 *are the private keys; or*
- $\mathsf{m}_1@z$, $\mathsf{m}_2@z$ *represent private keys of which* m_3, m_4 *are the public keys.*

The "content analysis" inference rule **(**$\vdash$**C)** then states that if an actor can derive m_1 and m_2 such that $(\mathsf{m}_1 \doteq \mathsf{m}_2) \Rightarrow (\mathsf{m}_3 \doteq \mathsf{m}_4)$, and he can derive a message with m_3 in it, he can also derive the message with m_3 replaced by m_4, and vice versa.

Example 3. Let $\mathcal{C}_a = \{\mathcal{H}(id, age)|_1^\eta, id|_2^\eta, age|_3^\eta\}$ be the set of messages known by actor a with $id \in \mathsf{I}$, $age \in \mathsf{D}$ such that $id|_1^\eta \doteq id|_2^\eta$ and $age|_1^\eta \doteq age|_3^\eta$. $\mathcal{C}_a \vdash \mathcal{H}(id, age)|_1^\eta$ holds, and by **(**$\vdash$**CC)**, **(**$\vdash$**CH)** we have $\mathcal{C}_a \vdash \mathcal{H}(id|_2^\eta, age|_3^\eta)$. From this, a knows that $id|_1^\eta \doteq id|_2^\eta$ (as well as $age|_1^\eta \doteq age|_3^\eta$). By **(**$\vdash$**C)** he can then deduce $id|_1^\eta$:

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

In the same way also $\mathcal{C}_a \vdash age|_1^\eta$ follows. □

There are two notable consequences of content analysis. First, if an actor knows a public/private key pair in one context (ζ, k) and just the public key in another context (η, l) then he can deduce the private key in (η, l). Second, an actor can link different profiles of the same entity if he sees that the profiles share an identifier (see §4).

The feasibility of implementing the content analysis rule follows from two observations. First, we can safely assume that content analysis rules are the final steps (from leaf to root) in a deduction tree, and that messages m_1, m_2 in **(**$\vdash$**C)** have been deduced without content analysis. Second, in **(**$\vdash$**C)**, $\mathsf{n}_1 \doteq \mathsf{n}_2$ holds. Thus, to decide whther a given message n_2 can be derived, one can first derive without using **(**$\vdash$**C)** all messages n_1 content equivalent to n_2, and then verify whether any n_1 can be transformed step-by-step to n_2 using **(**$\vdash$**C)**.

3.3 Deduction on Object vs Information Layer

Given context messages $\mathcal{C}_a$, one can perform object layer deduction and then apply σ to the result; or one can first apply σ to $\mathcal{C}_a$ and then perform information layer deduction. One proves easily that the first approach gives at least as much information as the second, i.e., object layer deduction is at least as expressive as information layer deduction:

Proposition 1. *Let* $\mathcal{C}_a \subset \mathcal{L}^c$*. Define* $\overline{\mathcal{C}_a}\sigma := \{x\sigma \mid \mathcal{C}_a \vdash x\}$*;* $\overline{\mathcal{C}_a^i} := \{x \mid \sigma(\mathcal{C}_a) \vDash x\}$*. Then,* $\overline{\mathcal{C}_a^i} \subset \overline{\mathcal{C}_a}\sigma$*. Conversely,* $\overline{\mathcal{C}_a^i} \supset \overline{\mathcal{C}_a}\sigma$ *holds for all* $\mathcal{C}_a$ *iff* τ *is injective on* $\mathcal{L}^i$*.*

Note that object layer deduction is strictly more expressive than information layer deduction when τ is not injective, i.e., when different pieces of information have the same contents. This condition reflects a significant difference between IdM and other settings: in IdM, it is likely to come across different pieces of information with the same contents, whereas in other settings the kind of information that is usually considered – nonces, keys, random values, etc. – can for the purposes of analysis be safely assumed to have unique contents.

4 Knowledge of Personal Information

In this section we define the *view* of an actor a, capturing his knowledge about personal information. There are two aspects to this knowledge. First, what information the actor knows, formalized by the set $\mathcal{O}_a^c \subset \mathcal{O}^c$ of *detectable* context items. Second, which context items he knows to represent information about the same entity, formalized by the $\leftrightarrow_a$ equivalence relation on $\mathcal{O}^c$ defining context items *associable* to each other.

An actor's view follows from his sets $\mathcal{C}_a \subset \mathcal{L}^c$, $\mathcal{E}_a^c \subset \mathcal{E}^c$ of known context messages and entities. Associations between context items follow from properties of both σ and τ. First, context items in one context are related, and so is the same entity in different contexts (properties 3, 4 of σ). Second, context identifiers with equal contents are equal (property 1 of τ). Thus, define $\leftrightarrow_a$ as the minimal equivalence relation on $\mathcal{O}^c$ such that:

- For all $e|_k^\eta, e|_l^\zeta \in \mathcal{E}^c$: $e|_k^\eta \leftrightarrow_a e|_l^\zeta$; for all $x|_k^\eta, y|_k^\eta \in \mathcal{O}^c$: $x|_k^\eta \leftrightarrow_a y|_k^\eta$
- If $\mathcal{C}_a \vdash \mathsf{m}_1, \mathcal{C}_a \vdash \mathsf{m}_2$, 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 $\mathsf{i}_1 \leftrightarrow_a \mathsf{i}_2$.

Detectability of items follows from our deductive system: $\mathcal{O}_a^c = \mathcal{E}_a^c \cup \mathsf{I}_a^c \cup \mathsf{D}_a^c$, where $\mathsf{D}_a^c = \{\mathsf{d} \in \mathsf{D}^c \mid \mathcal{C}_a \vdash \mathsf{d}\}$ and $\mathsf{I}_a^c = \{\mathsf{i} \in \mathsf{I}^c \mid \mathcal{C}_a \vdash \mathsf{i}\}$. One may expect that $e|_k^\eta \in \mathcal{E}_a^c$ and $e|_k^\eta \leftrightarrow_a i|_l^\chi$ imply $e|_l^\chi \in \mathcal{E}_a^c$, but, as can be seen later, we do need such a rule to define the view as $e|_k^\eta$ and $e|_k^\eta$ will be known by the actor to be equivalent anyway.

Note that actors may associate items which they can not detect. In fact, because of transitivity of $\leftrightarrow_a$, an actor knowing a relation between items he can not detect may help him to establish a relation between items he can detect:

Example 4. Consider a set $\mathcal{C}_a = \{\{E_{shakey|.}(id|_1), d|_1\}|^\eta, \{E_{shakey|.}(id|_1), d'|_1\}|^\chi\}$ of messages known by actor a, where $E_{shakey|.}(id|_1)|^\eta \doteq E_{shakey|.}(id|_1)|^\chi$. Then, $id|_1^\eta \leftrightarrow_a id|_1^\chi$ by condition 2 for $\leftrightarrow_a$ (even though the actor can detect neither context identifier). By condition 1 for $\leftrightarrow_a$ and transitivity, $d|_1^\eta \leftrightarrow_a d'|_1^\chi$ follows. □

We simplify the representation of an actor's knowledge by considering his *known equivalences* $\equiv_a$, defined as follows: $x \equiv_a y$ if $x, y \in \mathcal{O}_a^c$, $x \equiv y$ and $x \leftrightarrow_a y$.

Definition 3. *Let a be an actor with set of known context messages $\mathcal{C}_a$. Then, a's* view *is the structure $M_a^c = (\mathcal{E}_a^c / \equiv_a, \mathsf{I}_a^c / \equiv_a, \mathsf{D}_a^c / \equiv_a, \leftrightarrow_a / \equiv_a)$ with $\leftrightarrow_a / \equiv_a$ the canonical equivalence relation on $\mathcal{O}_a^c / \equiv_a$.*

5 Defining and Verifying Identity-Related Properties

In this section, we recap the identity-related properties defined in our previous work [12], adapted to the three-layer model (see Table 1). Identity-related properties with respect to an actor can be seen either as properties of a data item (i.e., on the information layer), or of a context data item representing that data item (i.e., on the object layer). For instance, anonymity of a context data item d with respect to an actor a means that d is not associable by a to a context entity. However, there might be another, equivalent, context data item that *can* be associated by a to a context entity.

We define identity-related properties for a data item d by considering the privacy properties holding for all context data items that represent it; for example, d is anonymous if all its representations are. Note that for complete identifiability of a data item d,

Table 1. Identity-related properties with respect to actor a, defined for a context data item d (middle column) and for a data item d (right column), where $[d] := \{\mathsf{d} \in \mathsf{D}^c \mid \mathsf{d}\sigma = d\}$

Property	Condition on $\mathsf{d} \in \mathsf{D}^c$	Condition on $d \in \mathcal{D}$
detectability (D)	$\mathsf{d} \in \mathsf{D}^c_a$	$\exists \mathsf{d} \in [d]$: d is D
undetectability (UD)	$\mathsf{d} \notin \mathsf{D}^c_a$	$\forall \mathsf{d} \in [d]$: d is UD
identifiability (I)	$\exists \mathsf{e} \in \mathcal{E}^c_a$ s.t. $\mathsf{d} \leftrightarrow^c_a \mathsf{e}$	$\exists \mathsf{d} \in [d]$: d is I
pseudo-identifiability (PI)	$\exists \mathsf{i} \in \mathsf{I}^c_a$ s.t. $\mathsf{d} \leftrightarrow^c_a \mathsf{i}$	$\exists \mathsf{d} \in [d]$: d is PI
complete identifiability (CI)	$\exists \mathsf{e} \in \mathcal{E}^c_a, \mathsf{i} \in \mathsf{I}^c_a$ s.t. $\mathsf{d} \leftrightarrow^c_a \mathsf{e} \wedge \mathsf{d} \leftrightarrow^c_a \mathsf{i}$	$\exists \mathsf{d} \in [d]$: d is CI
anonymity (A)	$\mathsf{d} \notin \mathsf{D}^c_a$, or $\forall \mathsf{e} \in \mathcal{E}^c_a : \mathsf{d} \not\leftrightarrow{}^c_a \mathsf{e}$	$\forall \mathsf{d} \in [d]$: d is A
pseudonymity (PA)	$\forall \mathsf{e} \in \mathcal{E}^c_a$ $\mathsf{d} \not\leftrightarrow{}^c_a \mathsf{e}$ and $\exists \mathsf{i} \in \mathsf{I}^c_a$ s.t. $\mathsf{d} \leftrightarrow^c_a \mathsf{i}$	$\forall \mathsf{d} \in [d]$: d is A $\wedge$ $\exists \mathsf{d} \in [d]$: d is PI
complete anonymity (CA)	$\mathsf{d} \notin \mathsf{D}^c_a$, or $\forall \mathsf{e} \in \mathcal{E}^c_a$ $\mathsf{d} \not\leftrightarrow{}^c_a \mathsf{e}$ and $\forall \mathsf{i} \in \mathsf{I}^c_a$ $d \not\leftrightarrow{}^c_a \mathsf{i}$	$\forall \mathsf{d} \in [d]$: d is A $\wedge d$ is not PI
linkability (L) (to d'/d')	$\mathsf{d} \leftrightarrow_a \mathsf{d}'$	$\exists \mathsf{d}' \in [d']$: $\mathsf{d} \leftrightarrow_a \mathsf{d}'$
linkability (UL) (to d'/d')	$\mathsf{d} \not\leftrightarrow_a \mathsf{d}'$	$\nexists \mathsf{d}' \in [d']$: $\mathsf{d} \leftrightarrow_a \mathsf{d}'$

we require that the *same* representation of d is both identifiable and pseudo-identifiable; the other properties are obvious. The method developed in the previous sections then allows one to verify identity-related properties in the following three steps:

- **Step 1**: Using the deductive system, determine the detectable context items.
- **Step 2**: Determine associable context items, and thus the actor view.
- **Step 3**: From the actor view, check which properties are satisfied.

6 Case Study: TAS^3 Attribute Aggregation

In this section, we demonstrate our approach by analyzing the TAS^3 attribute aggregation infrastructure [16]. We demonstrate how our approach can be used to verify whether the privacy properties for which it was designed do indeed hold. To be able to check for linkability between different executions, we analyze two executions of the protocol involving the same actors. The results also hold for more than two executions. The analysis leads to some recommendations for improvements to the system.

6.1 TAS^3 Attribute Aggregation

The TAS^3 project (`http://tas3.eu`) is a research project aiming to create an architecture for on-line services based on personal information. Here we focus on the TAS^3 attribute aggregation infrastructure, in which a service provider (SP) collects from different identity providers (IdPs) personal information about a user requesting a service. A main feature of the infrastructure is the linking service (LS), which links the different identifiers of the user at different IdPs, alleviating the need for global user identifiers.

The attribute aggregation infrastructure is described at high level in [2,15,16], and the concrete message formats are described in [17]. These message formats are based on open standards: notably, SAML 2.0 [18] and Liberty ID-WSF 2.0 [19].

The infrastructure aims to guarantee a number of privacy properties [2,16]. First, the SP wants "strong cryptographic evidence that each of the [attributes] does belong to the user who has initiated the session" (*P1*). Second, "none of the user's [IdPs should] know about any of the user's other ones" (*P2*). Third, "the [LS should] not know who the user is, or what identity attributes [he has]" (*P3*). Finally, "the [SP should not be able to] relate visits of the user together" (*P4*).

In our case study, we consider one user with attributes at two different identity providers (IdP1 and IdP2), who wishes to access a service from one SP twice. Thus, the same attribute aggregation process takes place twice. The process begins after IdP1 has authenticated the user. IdP1 informs the SP that the user has been authenticated, provides the SP with the value of the user's attribute at IdP1, and refers the SP to the LS. The SP contacts the LS, who refers him to IdP2. Finally, the SP requests and receives the value of the user's attribute at IdP2.

6.2 Formalization

Our formalization of attribute aggregation is depicted in Fig. 4. Fig. 4(a) shows the messages exchanged in an instance of the attribute aggregation protocol. Fig. 4(b) shows the information layer. The user has profiles at IdP1 and IdP2 consisting of one attribute and one identifier. Also, the LS shares an identifier of the user with each of the two IdPs. Each communicating party (SP, LS, IdP1, IdP2) has a private/public key pair and a public identifier. Finally, the protocol instances use four nonces in total.

Fig. 4(c) displays the actors' knowledge in the object layer before attribute aggregation. The LS, IdP1 and IdP2 know the aforementioned information about the user in a context corresponding to some entry in their respective databases: say $|^{\lambda}_{21}$, $|^{\iota}_{7}$, and $|^{\pi}_{3}$. They also know the public keys and identifiers of the other actors, and their own secret key, in contexts corresponding to their roles in the system: $|^{\pi}_{\mathrm{SP}}$, $|^{\pi}_{\mathrm{LS}}$, $|^{\pi}_{\mathrm{IdP1}}$, and $|^{\pi}_{\mathrm{IdP2}}$. The map σ linking these context items to information is straightforward. Fig. 4(d) shows the messages known by each actor after two instances of attribute aggregation. We assume that each actor learns only the messages that he sent or received.

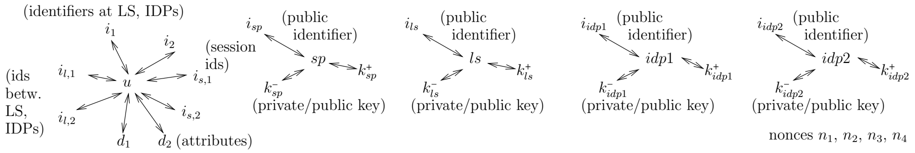
Finally, Fig. 4(e) formalizes the identity-related properties we previously introduced informally. Note that P2 and P3 are at the information layer, whereas P1 and P4, being about linking copies from different contexts, are at the object layer.

For the construction of our model, the high-level protocol descriptions from [2,16] were detailed enough. However, some lower-level aspects can be considered for extended analysis. First, the communication channels used in the protocol are all encrypted, which one could explicitly model. Second, in an implementation, the role of IdP2 would be performed by two logically different parties: a "discovery service" and an "attribute authority", so the communication would be more complex than we sketched. Third, the actual messages may contain information such as timestamps that ensure message portions from one context can not be re-used in another context.

Our formalization of the properties of the architecture differs slightly from their natural language descriptions in two ways. First, P1 mentions "strong cryptographic proof", suggesting that it is interesting to model the assurance an actor has about correctness of information he learns. In our previous work [12], we introduced notions of "provability" and "deniability" that could be used, but the present analysis method does not cover them. Second, P2 can have a stricter interpretation; that is, IdP1 should not even know whether or not the user has a profile at IdP2, and vice versa. However, in [15],

$$\begin{array}{rl}
IDP_1 \rightarrow SP & m_1 = S_{k^-|_{\mathrm{IDP}_1}}(i_{sess}|_{\mathrm{U}}, d_{idp1}|_{\mathrm{U}}, \{i|_{\mathrm{LS}}, E_{k^+|_{\mathrm{LS}}}(i_{idp1,ls}|_{\mathrm{U}}, n|.)\}) \\
SP \rightarrow LS & m_2 = \{E_{k^+|_{\mathrm{LS}}}(i_{idp1,ls}|_{\mathrm{U}}, n|.)\}), m_1\} \\
LS \rightarrow SP & m_3 = \{i|_{\mathrm{IDP}_2}, E_{k^+|_{\mathrm{IDP}_2}}(i_{idp2,ls}|_{\mathrm{U}}, n'|.)\} \\
SP \rightarrow IDP_2 & m_4 = \{E_{k^+|_{\mathrm{IDP}_2}}(i_{idp2,ls}|_{\mathrm{U}}, n'|.), m_1\} \\
IDP_2 \rightarrow SP & m_5 = S_{k^-|_{\mathrm{IDP}_2}}(i_{sess}|_{\mathrm{U}}, d_{idp2}|_{\mathrm{U}})
\end{array}$$

(a) Protocol description



(b) Information layer

$$\mathcal{C}^0_{sp} = \mathcal{C}_{pub} \cup \{k^-|^{\pi}_{\mathrm{SP}}\}, \mathcal{C}^0_{ls} = \mathcal{C}_{pub} \cup \{k^-|^{\pi}_{\mathrm{LS}}, i_p|^{\lambda}_{21}, i_{s,1}|^{\lambda}_{21}, i_{s,2}|^{\lambda}_{21}\},$$
$$\mathcal{C}^0_{idp1} = \mathcal{C}_{pub} \cup \{k^-|^{\pi}_{\mathrm{IDP}_1}, i_p|^{\iota}_{7}, i_s|^{\iota}_{7}, d|^{\iota}_{7}\}, \mathcal{C}^0_{idp2} = \mathcal{C}_{pub} \cup \{k^-|^{\pi}_{\mathrm{IDP}_2}, i_p|^{\iota'}_{2}, i_s|^{\iota'}_{2}, d|^{\iota'}_{2}\},$$
$$\text{with } \mathcal{C}_{pub} = \{k^+|^{\pi}_{\mathrm{SP}}, i|^{\pi}_{\mathrm{SP}}, k^+|^{\pi}_{\mathrm{LS}}, i|^{\pi}_{\mathrm{LS}}, k^+|^{\pi}_{\mathrm{IDP}_1}, i|^{\pi}_{\mathrm{IDP}_1}, k^+|^{\pi}_{\mathrm{IDP}_2}, i|^{\pi}_{\mathrm{IDP}_2}\}.$$

(c) Object layer: initial knowledge

$$\mathcal{C}_{sp} = \mathcal{C}^0_{sp} \cup \{\{m_1, m_2, m_3, m_4, m_5\}|^{\pi,1}, \{m_1, m_2, m_3, m_4, m_5\}|^{\pi,2}\},$$
$$\mathcal{C}_{ls} = \mathcal{C}^0_{ls} \cup \{\{m_2, m_3\}|^{\pi,1}, \{m_2, m_3\}|^{\pi,2}\}, \quad \mathcal{C}_{idp1} = \mathcal{C}^0_{idp1} \cup \{m_1|^{\pi,1}, m_1|^{\pi,2}\},$$
$$\mathcal{C}_{idp2} = \mathcal{C}^0_{idp2} \cup \{\{m_4, m_5\}|^{\pi,1}, \{m_4, m_5\}|^{\pi,2}\}$$

(d) Object layer: knowledge after two instances $(\pi, 1)$, $(\pi, 2)$ of attribute aggregation

- P1: copies of d_1, d_2 in same context detectable and linkable w.r.t. SP
- P2: i_2, d_2 undetectable w.r.t. IdP1; i_1, d_1 undetectable w.r.t. IdP2
- P3: i_1, i_2, d_1, d_2 undetectable, all items of interest related to user anonymous w.r.t. LS
- P4: copies of d_1, d_2 from different contexts unlinkable w.r.t. SP

(e) Goals for actor knowledge

Fig. 4. Formal model of TAS^3 attribute aggregation

it is specified that this interpretation does not hold for IdP2 w.r.t. IdP1. To capture this interpretation, in general, one would need to define an actor's knowledge about the knowledge of another actor, which is not possible in our model. Accordingly, our interpretation is really less strict: if the architecture does not satisfy our version of P2, then it also does not satisfy the strict version; however, the opposite implication does not hold.

6.3 Formal Analysis and Discussion

We follow the three steps outlined in Section 5 to check whether the properties in Fig. 4(e) hold in the formal model in Figs. 4(a)–4(d). Our results have been obtained using a Prolog implementation of the deductive system. First we check the properties about the SP's knowledge: P1 and P4. Step 1 gives $\mathcal{C}_{sp} \vdash d_{idp1}|^{\pi,1}_{\mathrm{U}}$ and $\mathcal{C}_{sp} \vdash d_{idp2}|^{\pi,1}_{\mathrm{U}}$, and step 2 gives $d_{idp1}|^{\pi,1}_{\mathrm{U}} \leftrightarrow_{sp} d_{idp2}|^{\pi,1}_{\mathrm{U}}$. Because $d_{idp1}|^{\pi,1}_{\mathrm{U}}\sigma = d_1$ and $d_{idp2}|^{\pi,1}_{\mathrm{U}}\sigma = d_2$, the copies of d_1, d_2 in $(\pi, 1)$ are detectable and linkable w.r.t. the SP. The same applies to the copies in $(\pi, 2)$. Thus, P1 holds. For P4, we need to check that items from different contexts are unlinkable w.r.t. SP, i.e., $d_{idp1}|_{\mathrm{U}}^{\pi,1} \leftrightarrow_{sp} d_{idp1}|_{\mathrm{U}}^{\pi,2}$ can not be derived in step 2. Indeed the link cannot be made, so TAS^3 attribute aggregation satisfies P4. Note that this conclusion crucially depends on the nonces being different between protocol instances. Note also that these properties depend on linking and distinguishing information instances and so they cannot be verified using standard deductive systems.

On the other hand, P2 and P3 do not hold: both LS and IdP2 can detect the objects $d_{idp1}|_{\mathrm{U}}^{\pi,*}$ (with $* \in \{1, 2\}$) representing the information d_1. This is due to message m_1, representing the authentication assertion signed by IdP1, being included in the messages from the SP to the LS and IdP2. However, undetectability of i_1, i_2, and d_2 and anonymity of these items w.r.t. the LS do hold. As in [15], we see that the stricter interpretation of P2 that we discussed earlier does not hold: indeed, IdP2 receives an authentication assertion about the user which it knows it has been signed by IdP1. Finally, note that all parties involved in the protocol learn the session identifiers $i_{s,1}$ and $i_{s,2}$ in the process. In particular, if IdP1 and IdP2 collude, then from their known messages they can link their user profiles — again a conclusion of studying relations between personal information that standard deductive systems cannot express.

Our analysis leads to two recommendations on how privacy in TAS^3 attribute aggregation may be improved. First, the SP should not forward the attribute d_1 from IdP1 to the LS and IdP2. However, implementing this is difficult. Indeed, according to the TAS^3 attribute aggregation requirements, the LS and IdP2 should receive a signed authentication assertion from IdP1 to be sure that the user did actually authenticate. In the standards used in TAS^3, the attribute value is part of that authentication assertion. Therefore, a mechanism is desired that allows IdP1 to prove that the user has authenticated without disclosing attribute values. Second, the problem of collision between IdP1 and IdP2 should be avoided by not having a shared identifier between IdP1 and IdP2, but this requires IdP2 to trust the LS that the user has indeed been authenticated.

7 Conclusion and Future Work

In this paper, we considered privacy in IdM by presenting a novel method for privacy analysis of communication protocols. We presented a three-layer model of personal information and showed that it allows for an accurate representation of an actor's knowledge. We showed how to reason about this model using deductive methods, and how to check which privacy properties hold after communication. We demonstrated the feasibility of our approach by a) showing that existing deductive systems can be adapted to our approach; b) proving that such an adaptation does not reduce the expressiveness of the deductive system; and c) performing an case study which made it possible to identify a number of privacy issues in the design of an existing IdM architecture.

This work provides several interesting directions for future work. First, we aim to integrate our three-layer model and deductive system into a state transition system approach. This makes it possible to fully automate the protocol verification, and provides opportunities for the development of tooling. The ability to model false information and probabilistic knowledge of links provides an interesting connection to record linkage theory [20]. Namely, it raises the question whether an entity can be identified (almost) uniquely from a profile with data items that by themselves are not identifying. Another extension to the model is to consider provability of links between pieces of information. The signed authentication assertion from TAS^3 attribute aggregate is an example application for this; electronic payment systems are another. Finally, we are analyzing a number of IdM systems and modeling additional cryptographic primitives they use.

References

1. Sommer, D. (ed.): PRIME Architecture V3. Version 1.0, `http://www.prime-project.eu/`
2. Kellomäki, S. (ed.): D2.1 - TAS^3 Architecture. Version 17, `http://tas3.eu/`
3. Scavo, T., Cantor, S. (eds.): Shibboleth Architecture: Technical Overview. Working Draft 02, `http://shibboleth.internet2.edu/shibboleth-documents.html`
4. Abadi, M., Gordon, A.D.: A calculus for cryptographic protocols: the spi calculus. In: Proc. of CCS 1997, pp. 36–47. ACM (1997)
5. Burrows, M., Abadi, M., Needham, R.: A logic of authentication. ACM Trans. Comput. Syst. 8, 18–36 (1990)
6. Meadows, C.: Formal methods for cryptographic protocol analysis: emerging issues and trends. IEEE Journal on Selected Areas in Comm. 21(1), 44–54 (2003)
7. Paulson, L.C.: The Inductive Approach to Verifying Cryptographic Protocols. Journal of Computer Security 6(1-2), 85–128 (1998)
8. Bella, G., Paulson, L.: Kerberos Version IV: Inductive Analysis of the Secrecy Goals. In: Quisquater, J.-J., Deswarte, Y., Meadows, C., Gollmann, D. (eds.) ESORICS 1998. LNCS, vol. 1485, pp. 361–375. Springer, Heidelberg (1998)
9. Delaune, S., Ryan, M., Smyth, B.: Automatic verification of privacy properties in the applied pi calculus. In: Trust Management II. IFIP AICT, vol. 263, pp. 263–278. Springer, Heidelberg (2008)
10. Aziz, B., Hamilton, G.: A Privacy Analysis for the π-calculus: The Denotational Approach. In: Proc. of SAVE 2002, Copenhagen, Denmark (July 2002)
11. Brusò, M., Chatzikokolakis, K., den Hartog, J.: Formal Verification of Privacy for RFID Systems. In: Proc. of CSFW 2010, pp. 75–88. IEEE (2010)
12. Veeningen, M., de Weger, B., Zannone, N.: Modeling Identity-Related Properties and Their Privacy Strength. In: Degano, P., Etalle, S., Guttman, J. (eds.) FAST 2010. LNCS, vol. 6561, pp. 126–140. Springer, Heidelberg (2011)
13. Clarke, E., Jha, S., Marrero, W.: Using state space exploration and a natural deduction style message derivation engine to verify security protocols. In: Proc. of ICPCM 1998, pp. 86–106. Chapman & Hall, Ltd., Boca Raton (1998)
14. Ramsdell, B., Turner, S.: Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2: Message Specification. RFC 5751 (2010)
15. Chadwick, D., Inman, G.: Attribute Aggregation in Federated Identity Management. IEEE Computer 42(5), 33–40 (2009)
16. Chadwick, D. (ed.): Design of Identity Management, Authentication and Authorization Infrastructure. Version 2.1.1, `http://tas3.eu/`
17. TAS^3 Protocols, API, and Concrete Architecture. Version 10, `http://tas3.eu/`
18. Cantor, S., Kemp, K., Philpott, R., Maler, E. (eds.): Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard, (March 15, 2005), `http://saml.xml.org/saml-specifications`
19. Hodges, J., Kemp, K., Aarts, R., Whitehead, G., Madsen, P. (eds.): Liberty ID-WSF SOAP Binding Specification. Version 2.0, `http://projectliberty.org/`
20. Fellegi, I., Sunter, A.: A Theory for Record Linkage. Journal of the American Statistical Association 64(328), 1183–1210 (1969)