Symbolic Privacy Analysis through Linkability and Detectability

Meilof Veeningen, Benne De Weger, Nicola Zannone

Symbolic Privacy Analysis through Linkability and Detectability

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. More and more personal information is exchanged on-line using communication protocols. This makes it increasingly important that such protocols satisfy privacy by data minimisation. Formal methods have been used to verify privacy properties of protocols; but so far, mostly in an ad-hoc way. In previous work, we provided general definitions for the fundamental privacy concepts of linkability and detectability. However, this approach is only able to verify privacy properties for given protocol instances. In this work, by generalising the approach, we formally analyse privacy of communication protocols independently from any instance. We implement the model; identify its assumptions by relating it to the instantiated model; and show how to visualise results. To demonstrate our approach, we analyse privacy in Identity Mixer.

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. There have been numerous reports of such information being used for secondary purposes [17], or being stolen and abused by third parties [15]. Legislation (e.g., EU Directive 95/46/EC, HIPAA) attempts to reduce these risks by demanding organisations to collect and store the minimal amount of information they need: the *data minimisation* principle. An important factor in achieving it is the use of protocols to exchange personal information that are *privacy-enhancing* [9], i.e., protocols that use cryptographic primitives to ensure that their participants learn as little information as possible, and that they have as little ability as possible to link information from different sources into one profile.

However, it is hard to precisely and accurately assess the privacy offered by such privacy-enhancing protocols. Surveys analyse privacy in areas such as e-health [13, 16] or identity management [11, 10], but these analyses are performed in an informal and high-level (and thus, possibly subjective) way. Formal methods have been successfully applied to protocol verification [12]. However, traditionally, they focus mostly on secrecy of isolated pieces of information with respect to a malicious outsider, whereas privacy also concerns the building of profiles of personal information by authorised insiders who combine different pieces of information. A recent body of works [6–8] extends these formal methods to analyse links between different pieces of information; however, properties are mostly defined in an ad-hoc fashion for particular protocols.

In [19, 20], we presented a general formal model for privacy analysis of communication protocols. The model captures privacy, irrespective of any particular protocol, as

two fundamental concepts: detectability (what personal information is known) and linkability (what personal information is known to be about the same person). By making types of information (non-personal information, identifiers, attributes, etc.) an integral part of our model, we get one single representation of the knowledge of actors, in which different privacy properties can be defined and checked [19], and data minimisation assessed [18]. However, detectability and linkability are verified only for a set of protocol instances with given information, without guarantee of generalisation.

In this work, we extend and generalise the approach above to enable symbolic privacy analysis of communication protocols. By analysing a symbolic model of protocols rather than an instantiated model of protocol instances, we draw privacy conclusions that apply to any particular scenario. Specifically:

- We derive *constraints* describing exactly, for any interaction between legitimate actors, under which conditions detectability and linkability hold (we do not consider malicious actors). We provide a tool to compure these constraints automatically;
- We present the *constraints graph*, a visual representation of relevant constraints;
- By relating symbolic conclusions to our previous model [19, 20], we describe precisely under what assumptions on instantiations our conclusions hold;
- We demonstrate feasibility of our approach by a privacy analysis of Identity Mixer.

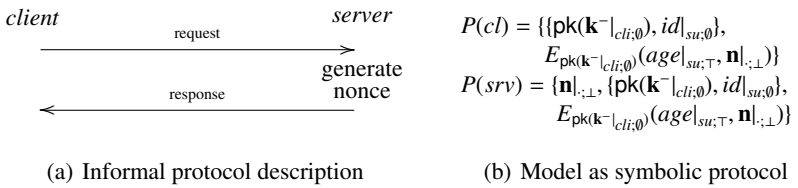
This paper is structured as follows. We survey related work (§2); model protocols and their instantiations (§3); model reasoning on instantiations (§4) and generalise this to symbolic protocols (§5); apply our methods to Identity Mixer (§6); and conclude (§7).

2 Related Work

Formal methods are widely used as a tool for the analysis of security in communication protocols [2, 5, 12, 14]. Formal methods generally rely on two basic ideas: the Dolev-Yao attacker model and state exploration. The Dolev-Yao model describes an attacker who can intercept and manipulate communicated messages built using cryptographic primitives. Deductive systems (e.g., [5]) or equational theories (e.g., [2]) describe how he can change messages or derive secrets from them. State exploration techniques are then used to analyse all possible states that can be reached by a system of interacting actors in the face of such an attacker. Protocols are commonly modelled using process algebras (e.g., [2]); alternative approaches exist, e.g., using induction [14].

A recent body of works uses process algebraic models to analyse linkability in, e.g., electronic toll collection [6], eHealth [7], and e-voting [8]. Linkability is expressed in terms of "experiments": pairs of scenarios that should be indistinguishable to an attacker. However, these experiments are usually specific to the particular protocol being verified, making it hard to compare different systems. Also, each experiment looks at one aspect of the protocol in isolation, making it hard to ensure the set of experiments is representative of all possible system instantiations. Finally, the most general experiments (in which infinitely many actors simultaneously perform infinitely many protocol runs) are often too complicated for automated analysis, so simplifications are needed.

Recent work [3] proposes to define and verify linkability using the inductive method [14]. The use of interactive proofs potentially allows more general experiments to be



(a) Informal protocol description (b) Model as symbolic protocol

Fig. 1. A simple protocol: informal description (left), model as a symbolic protocol (right)

analysed. However, in terms of analysed properties, this approach is even less general than the process algebraic one: their definition depends not just on the protocol but also on the format of its messages; it is not clear to what extent this can be generalised.

Our approach does not suffer from the disadvantages mentioned above. We provide privacy definitions that are generic, expressible in a single model, and feasibly evaluatable, regardless of the number of actors in a scenario. However, in contrast to the above works, we do not capture outside attackers.

3 Symbolic Protocol Model and Instantiation

In this section, we present our symbolic model of protocols for exchanging personal information (§3.1). Next, by modelling instantiations of these symbolic models, we express precisely to what real-life scenarios our model applies (§3.2).

3.1 Symbolic Protocol Model

We present a model of communication protocols that is suitable for analysing their privacy. A communication protocol describes formats for messages (typically using cryptographic primitives), and rules for exchanging them. In addition, the protocol assigns "types" to the pieces of information to restrict the contents they may have (e.g., a nonce may not be re-used, an identifier should uniquely identify its data subject).

Example 1. Consider a protocol between a client and a server, following the structure of Figure 1(a). First, the client sends a request to the server containing her public key and the identifier of a subject. The server generates a nonce, and responds with an asymmetric encryption with the client's public key of the subject's age and the nonce. The private/public key pair should be randomly generated; the identifier and age should refer to the same subject; the identifier shoud uniquely identify it; the nonce should be different in every protocol instance. □

For our purposes, each protocol involves a fixed number of *roles*. The role defines the messages that are sent and received by an actor performing that role, and the pieces of information that are generated by that actor (e.g., nonces). For instance, Example 1 has roles client *cl* and server *srv*. A *profile* represents a data subject whose personal information is exchanged. It can be a role, or it can represent an entity not involved in the protocol. For instance, Example 1 has profiles client *cl*, server *srv*, and subject *su*.

	Type of information	Random	Non-Random
Personal Information	- global identifier of A	$\mathbf{v}\|_{A;\emptyset}$	$v\|_{A;\emptyset}$
	- local identifier of A with respect to B	-	$v\|_{A;B}$
	- data item (non-identifying) of A	-	$v\|_{A;\top}$
Non-Personal Information	- instance-specific	$\mathbf{v}\|_{\cdot;\bot}$	-
	- non-instance-specific	$\mathbf{v}\|_{\cdot;\top}$	$v\|_{\cdot;\top}$

Fig. 2. Types of symbolic items and their notation (v a variable, A,B sequences of profiles)

Pieces of information occurring in protocol messages are represented by *symbolic items* $v|_{A;B}$ from set $\mathfrak{P}$. *Variable* v describes the piece of information. *Topic* A is an ordered sequence of profiles whom the piece of information is about, or $\cdot$ for non-personal information. For personal information (i.e., $A \neq \cdot$), *scope* B may be $\emptyset$, $\top$, or a sequence of profiles. *Global identifiers* ($B = \emptyset$) uniquely identify the entities represented by the profiles A. For instance, the identifier of the data subject in Example 1 is modelled $id|_{su;\emptyset}$. *Data items* ($B = \top$) do not uniquely identify their data subjects; for instance, the age $age|_{su;\top}$ of the subject in Example 1. *Local identifiers* ($B =$ a sequence of profiles) identify the entities in A only with respect to the the entities in B, e.g., an identifier of a user with respect to an identity provider. (Usually, A has length 1; however, e.g., A has length 2 for a shared key between two entities.) Non-personal information ($A = \cdot$) has *scope* $B \in \{\top, \bot\}$. *Instance-specific information* $v|_{\cdot;\bot}$ only occurs in one particular protocol instance, e.g., a nonce. *Non-instance specific information* $v|_{\cdot;\top}$ can occur in multiple instances, possibly of different protocols; e.g., a transaction date.

Apart from the type of information, we also indicate whether it is randomly generated. Because contents of randomly generated information cannot be guessed, this information can be used to hide other information (e.g., encryption keys, nonces). Random information is denoted by boldfaced variables; for instance, the private key of the client in Example 1 is modelled $\mathbf{k}^-|_{cli;\emptyset}$. Figure 2 shows the notation for different types of information. Note that some combinations are not allowed; for instance, a data item cannot be randomly generated because then it would in fact identify its data subject.

Messages built from these symbolic items using cryptographic primitives are represented by the set $\mathfrak{L}$ of *symbolic messages*. Formally, $\mathfrak{L}$ is a language that is built inductively from symbolic items $\mathfrak{P}$ using cryptographic primitives. For instance, private/public key pairs can be modelled by defining the public key corresponding to private key $\mathfrak{k}^-$ as $\mathsf{pk}(\mathfrak{k}^-) \in \mathfrak{L}$. Asymmetric encryption is modelled by defining the encryption of plaintext $\mathfrak{m}$ under public key $\mathsf{pk}(\mathfrak{k}^-)$ as $E_{\mathsf{pk}(\mathfrak{k}^-)}(\mathfrak{m}) \in \mathfrak{L}$. In general, primitives depend on the protocol, but we define one "standard" primitive: the list. A list (i.e., concatenation) of messages $\mathfrak{m}_1, ..., \mathfrak{m}_k$, $k \geq 0$ is denoted $\{\mathfrak{m}_1, ..., \mathfrak{m}_k\} \in \mathfrak{L}$. For instance, the messages in Example 1 are $\{\mathsf{pk}(\mathbf{k}^-|_{cli;\emptyset}), id|_{su;\emptyset}\}$ and $E_{\mathsf{pk}(\mathbf{k}^-|_{cli;\emptyset})}(\{age|_{su;\top}, \mathbf{n}|_{\cdot;\bot}\})\}$.

Apart from the fixed-length lists above, we also allow variable-length lists of items with a common format. A variable-length list of symbolic items is denoted $\{\mathfrak{v}\}_F$. Here, the symbolic item $\mathfrak{v}$ identifies the type of item (e.g., "attribute"); the *family* F identifies which set of items of that type is taken (e.g., *all*="all known items", *pub*="all public items", *dem*="all demographic information"). For instance, a variation of the second message in our example protocol where the age in the second message is replaced by

a list of all known attributes is modelled $E_{\mathsf{pk}(\mathbf{k}^-|_{cli;\emptyset})}(\{\{d|_{su;\top}\}_{all}, \mathbf{n}|_{\cdot;\perp}\})$. The individual elements of $\{\mathfrak{v}\}_F$ are denoted $\{\mathfrak{v}\}_F@k$ for $k = 1, 2, \ldots$; which of these items exist depends on the protocol instance. Variable-length lists containing cryptographic primitives are also allowed. Their items are defined piecewise, e.g., $\{E_{\mathsf{pk}(\mathbf{k}^-|_{cli;\emptyset})}(\{d|_{su;\top}, \mathbf{n}|_{\cdot;\perp}\})\}_{all}@k = E_{\{\mathsf{pk}(\mathbf{k}^-|_{cli;\emptyset})\}_{all}@k}(\{\{d|_{su;\top}\}_{all}@k, \{\mathbf{n}|_{\cdot;\perp}\}_{all}@k\})$ for $k = 1, 2, ...$

A *symbolic protocol* captures, for each role in the protocol, what messages are sent, received, and generated by the actor performing that role. In our framework, neither the order of these messages matters, nor whether they were sent, received, or generated. Thus, we simply assign a set of symbolic messages to each role:

Definition 1. *A* symbolic protocol $\mathfrak{Pr}$ *between roles* $r_1, ..., r_k$ *is a collection of sets* $\mathfrak{Pr}(r_i) \subset \mathfrak{L}$ *of symbolic messages, where each set* $\mathfrak{Pr}(r_i)$ *contains all messages sent, received and generated by the actor performing the role* r_i *in a full run the protocol.*

Example 2. The protocol described informally in Example 1 is formalised as the symbolic protocol P between client *cl* and server *srv* shown in Figure 1(b). □

3.2 Instantiated Model

An *instantiated model* captures personal information exchanged in a scenario. A scenario encompasses a number of instances of symbolic protocols; each instance is referred to by a *domain*. In each domain π, symbolic items $v|_*$, $\mathbf{v}|_*$ are instantiated to *context items* $v|^\pi_*$, $\mathbf{v}|^\pi_*$ representing actual pieces of information with actual contents (where $*$ is any topic and scope). Context item $v|^\pi_*$ is called the *instantiation* of $v|_*$ in π.

The structure of messages naturally extends to the instantiated model. Previously, we defined the set $\mathfrak{L}$ of symbolic messages by induction from symbolic items. The set $\mathcal{L}^c$ of *context messages* is defined in the same way from context items. The *instantiation* of symbolic message $\mathfrak{m}$ in domain π, denoted $\mathfrak{m}|^\pi$, is obtained by instantiating its symbolic items, e.g., $E_{\mathsf{pk}(\mathbf{k}^-|_{cli;\emptyset})}(age|_{su;\top}, \mathbf{n}|_{\cdot;\perp})|^\pi = E_{\mathsf{pk}(\mathbf{k}^-|^\pi_{cli;\emptyset})}(age|^\pi_{su;\top}, \mathbf{n}|^\pi_{\cdot;\perp})$. Variable-length lists are instantiated to normal lists, e.g. if $\{v|^\pi_p\}_I@1, \ldots, \{v|^\pi_p\}_I@k$ are defined, then $\{v|_p\}_I|^\pi = \{\{v|^\pi_p\}_I@1, \ldots, \{v|^\pi_p\}_I@k\}$. The set $\mathcal{L}^c$ is defined up to list nesting, e.g., $\{\mathsf{m}_1, \{\mathsf{m}_2, \mathsf{m}_3\}\}$ and $\{\mathsf{m}_1, \mathsf{m}_2, \mathsf{m}_3\}$ represent the same context message, as do $\{\mathsf{m}\}$ and m.

The instantiated model captures contents of context messages as well as personal relations between the entities they describe. The function $\phi(\mathsf{m})$ models the bitstring contents of context message m, e.g. $\phi(age|^\pi_{su;\top}) =$ '18'. If $\phi(\mathsf{m}) = \phi(\mathsf{n})$, then we call m and n *content equivalent*, denoted $\mathsf{m} \doteq \mathsf{n}$. The entity represented by profile p in domain π is referred to by the *context* $*|^\pi_p$. The $\leftrightarrow$ equivalence relation indicates which contexts belong to the same entity; e.g., $*|^\pi_{su} \leftrightarrow *|^\kappa_{su}$ expresses that the subjects in protocol instances π and κ are the same; $*|^\pi_{su}$ and $*|^\kappa_{su}$ are called *related*. The following definition details our assumptions on contents of messages and relations between contexts:

Definition 2. *An* instantiated model *is a tuple* $(\mathsf{P}^c, \phi, \leftrightarrow)$*, where:*

- P^c *is a set of context items; for any variable-length lists* $\{v|_*\}_F$, $\{w|_*\}_F$ *in domain* π*, the sets of* k *such that* $\{v|^\pi_*\}_F@k \in \mathsf{P}^c$ *and* $\{w|^\pi_*\}_F@k \in \mathsf{P}^c$ *are equal;*
- ϕ *maps context messages built from context items* $\in \mathsf{P}^c$ *to bitstrings* $\in \Sigma^*$ *such that:*
  - *If* $\phi(y) = \phi(\mathbf{v}|^\pi_{A;\emptyset})$*, then for some* κ, A'*,* $y = \mathbf{v}|^\kappa_{A';\emptyset}$*;* A, A' *have the same length;*

  - *If* $\phi(y) = \phi(\mathbf{v}|^{\pi}_{:,\bot})$*, then* $y = \mathbf{v}|^{\pi}_{:,\bot}$*;*
  - *If* $\phi(y) = \phi(\mathbf{v}|^{\pi}_{:,\top})$*, then for some* κ*,* $y = \mathbf{v}|^{\kappa}_{:,\top}$*;*
  - $\phi(E_{\mathsf{a}}(\mathsf{b})) = \phi(y)$ *if and only if* $y = E_{\mathsf{c}}(\mathsf{d})$*,* $\phi(\mathsf{a}) = \phi(\mathsf{c})$*, and* $\phi(\mathsf{b}) = \phi(\mathsf{d})$*; and similarly for all other primitives.*
- $\leftrightarrow$ *is an equivalence relation on contexts* $*|^{\pi}_{p}$ *occurring in* P^c *such that:*
  - *If* $\phi(\mathbf{id}|^{\pi}_{a_1,\ldots,a_k;\emptyset}) = \phi(\mathbf{id}|^{\kappa}_{b_1,\ldots,b_k;\emptyset})$*, then* $*|^{\pi}_{a_i} \leftrightarrow *|^{\kappa}_{b_i}$ *for* $i = 1,\ldots,k$*;*
  - *If* $\phi(id|^{\pi}_{a_1,\ldots,a_k;\emptyset}) = \phi(id|^{\kappa}_{b_1,\ldots,b_k;\emptyset})$*, then* $*|^{\pi}_{a_i} \leftrightarrow *|^{\kappa}_{b_i}$ *for* $i = 1,\ldots,k$*;*
  - *If* $\phi(id|^{\pi}_{a_1,\ldots,a_k;c_1,\ldots,c_l}) = \phi(id|^{\kappa}_{b_1,\ldots,b_k;d_1,\ldots d_l})$ *and* $*|^{\pi}_{c_i} \leftrightarrow *|^{\kappa}_{d_i}$ *for* $i = 1,\ldots,l$*; then* $*|^{\pi}_{a_i} \leftrightarrow *|^{\kappa}_{b_i}$ *for* $i = 1,\ldots,k$*.*

Domain π instantiates *symbolic protocol* $\mathfrak{Pr}$ *in instantiated model* $(\mathsf{P}^c, \phi, \leftrightarrow)$ *if* $\mathfrak{p}|^{\pi} \in \mathsf{P}^c$ *for all symbolic items* $\mathfrak{p}$ *occurring in messages in* $\mathfrak{Pr}$*.*

The restrictions on ϕ formalise the assumptions we make on random data and cryptographic primitives. The first three restrictions capture the assumption that the same random value is never generated twice: thus, a context item representing a random value can only be content equivalent to another context item of the same type with the same variable. The fourth restriction captures two assumptions on cryptographic primitives. The "if" part of the restriction expresses *determinism*, namely, given the same inputs, primitives always give the same output. Randomness should be modelled explicitly, e.g., as part of the plaintext in a non-deterministic encryption. The "only if" part expresses *structural equivalence*, namely, the contents of differently-constructed messages can never clash. That is, $E_{\mathsf{a}}(\mathsf{b})$ cannot be content equivalent to $E_{\mathsf{c}}(\mathsf{d})$ if $\phi(\mathsf{a}) \neq \phi(\mathsf{c})$ or $\phi(\mathsf{b}) \neq \phi(\mathsf{d})$; or to context items $\in \mathsf{P}^c$ or messages representing other primitives.

The restrictions on $\leftrightarrow$ formalise the uniqueness of identifiers. The first two restrictions state that global identifiers with the same contents should have related contexts. The third one states that local identifiers with the same contents should have related contexts if they are with respect to the same actors. (Because topic A of symbolic item $v|_{A;B}$ is a sequence, for instance, $\phi(id|^{\pi}_{a,b;\emptyset}) \neq \phi(id|^{\pi}_{b,a;\emptyset})$. If this is not desired, e.g., for some types of shared keys, a slight adaptation of the model is needed.)

4 Actor Knowledge and Reasoning in Instantiated Models

In this section, we formalise knowledge of personal information in instantiated models. Knowledge bases represent the knowledge of (coalitions of) actors (§4.1). Derivability expresses what messages actors can learn from a knowledge base using cryptographic operations (§4.2). Detectability and linkability express what sets of personal information they can compile about data subjects (§4.3). This model was presented in [19, 20].

4.1 Knowledge Base

The *knowledge base* of actor a captures the knowledge he obtains from his involvement in communication protocols. It is modelled by a set C_a of context messages. Fix an instantiated model $(\mathsf{P}^c, \phi, \leftrightarrow)$. If domain π_i instantiates protocol $\mathfrak{Pr}_i$ in $(\mathsf{P}^c, \phi, \leftrightarrow)$, and actor a performs role r_i in the protocol instance, then this contributes set $C_a|^{\pi_i} := \mathfrak{Pr}_i(r_i)|^{\pi_i}$ of context messages to his knowledge base; (C_a, π_i) is an *instantiation* of

$\mathfrak{Pr}_i(r_i)$. Knowledge base C_a is the union $C_a|^{\pi_1} \cup \ldots \cup C_a|^{\pi_k}$ of such contributions. We assume that C_a contains all relevant knowledge of a, including databases of personal information, keys, etc. The knowledge base of a coalition of actors is the union of the knowledge bases of the individual actors. In particular, if different actors in the coalition have performed different roles $r_1, \ldots, r_k$ in a single instance π_i of $\mathfrak{Pr}_i$, this contributes set $\mathfrak{Pr}_i(r_1) \cup \ldots \cup \mathfrak{Pr}_i(r_k)|^{\pi_i}$ to the knowledge base of the coalition.

4.2 Derivability

Cryptographic operations are captured by *construction* and *elimination* rules for messages representing cryptographic primitives. Construction, denoted $\mathsf{n}{\leftarrow}\mathsf{m}_1, \ldots, \mathsf{m}_k$, models how an actor can construct a message from its parts. Here, n is the context message that is constructed, and m_i are the context messages from which it is constructed. For instance, the rule for asymmetric encryption is $E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m}){\leftarrow}\mathsf{m}, \mathsf{pk}(\mathsf{k}^-)$, meaning that an encryption is constructed from the plaintext and public key. (Usually, primitives have one construction rule; the signature scheme used in our case study has two.)

Elimination, denoted with $\mathsf{n} \xrightarrow{\mathsf{req}_1,\ldots,\mathsf{req}_k} \mathsf{m}$, models how parts can be deconstructed from a message. Here, n is the message, m is the part to be deconstructed from it, and req_i are contents required to perform deconstruction. For instance, for asymmetric encryption we have $E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m}) \xrightarrow{\phi(\mathsf{k}^-)} \mathsf{m}$, meaning that if an actor knows encryption $E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m})$ and the contents of private key k^-, then he can learn message m. The actor only needs to know the contents of the key, not its context: that is, he can try out any private key he knows on $E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m})$: if he happens to use the correct one, he learns m.

Elimination rules include *testing rules*: rules for learning a new context message whose contents were already known in another context. Firstly, testing rules model the assumption of "visible failure" of cryptographic operations. For instance, testing rule $E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m}) \xrightarrow{\phi(\mathsf{k}^-)} \mathsf{k}^-$ states that by trying out a key on encryption $E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m})$ that happens to have the right contents, an actor learns the key k^- in the context of the encryption. Secondly, they model the possibility to reconstruct messages. By determinism, if an actor knows message n such that $\mathsf{n}{\leftarrow}\mathsf{m}_1, \ldots, \mathsf{m}_k$, and he knows the contents of all m_i, then by reconstructing message n he also learns each m_i. For instance, construction rule $E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m}){\leftarrow}\mathsf{m}, \mathsf{pk}(\mathsf{k}^-)$ for asymmetric encryption gives testing rules $E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m}) \xrightarrow{\phi(\mathsf{m}),\phi(\mathsf{pk}(\mathsf{k}^-))} \mathsf{m}$ and $E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m}) \xrightarrow{\phi(\mathsf{m}),\phi(\mathsf{pk}(\mathsf{k}^-))} \mathsf{pk}(\mathsf{k}^-)$.

The *derivability* relation $\vdash$ captures how actors can determine messages by repeatedly applying cryptographic operations. Its definition (Figure 3) consists of two parts: construction and elimination rules for the primitives used in a particular protocol (Figure 3(a) shows the primitives we use in our examples), and a deductive system for $\vdash$ (Figure 3(b)) that is the same for all protocols.

Definition 3. *Let C_a be a set of context messages, and m a context message. We say that m is* derivable *from C_a, denoted $C_a \vdash \mathsf{m}$, if the conclusion $C_a \vdash \mathsf{m}$ follows from the deductive system in Figure 3(b).*

Messages can be derived only by elimination steps ($\vdash\mathbf{E}$); however, contents that are needed for elimination steps may be constructed ($\vdash^+\mathbf{C}$). (We capture construction by the auxilliary symbol $\vdash^+$ which is only used to define $\vdash$.) Because we are only interested in deriving atomic pieces of personal information, this restricted definition suffices.

$$\textbf{(Private/public key)}\quad \mathsf{pk}(\mathsf{k}^-)\leftarrow\mathsf{k}^-\quad \mathsf{pk}(\mathsf{k}^-)\xrightarrow{\phi(\mathsf{k}^-)}\mathsf{k}^-$$

$$\textbf{(Asymmetric Encryption)}\quad E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m})\leftarrow\mathsf{m},\mathsf{pk}(\mathsf{k}^-)\quad E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m})\xrightarrow{\phi(\mathsf{k}^-)}\mathsf{m}$$

$$E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m})\xrightarrow{\phi(\mathsf{k}^-)}\mathsf{k}^-\quad E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m})\xrightarrow{\phi(\mathsf{m}),\phi(\mathsf{pk}(\mathsf{k}^-))}\mathsf{m}\quad E_{\mathsf{pk}(\mathsf{k}^-)}(\mathsf{m})\xrightarrow{\phi(\mathsf{m}),\phi(\mathsf{pk}(\mathsf{k}^-))}\mathsf{pk}(\mathsf{k}^-)$$

$$\textbf{(List)}\quad \{\mathsf{m}_1,\mathsf{m}_2,\mathsf{m}_3\}\leftarrow\mathsf{m}_1,\mathsf{m}_2,\mathsf{m}_3\quad \{\mathsf{m}_1,\mathsf{m}_2,\mathsf{m}_3\}\rightarrow\mathsf{m}_i\ (i=1,\ldots,k)$$

(a) Construction and elimination rules for asymmetric encryption and lists

$$\frac{}{C_a\vdash\mathsf{m}}\ (\mathsf{m}\in C_a)\quad(\vdash\mathbf{0})\qquad \frac{C_a\vdash\mathsf{m}\quad C_a\vdash^+\mathsf{m}_1\quad\ldots\quad C_a\vdash^+\mathsf{m}_k}{C_a\vdash\mathsf{n}}\ (\mathsf{m}\xrightarrow{\mathsf{r}_1,\ldots,\mathsf{r}_k}\mathsf{n},\forall i{:}\phi(\mathsf{m}_i)=\mathsf{r}_i)\quad(\vdash\mathbf{E})$$

$$\frac{C_a\vdash\mathsf{m}}{C_a\vdash^+\mathsf{m}}\ (\vdash^+\mathbf{0})\qquad \frac{C_a\vdash\mathsf{m}_1\quad\ldots\quad C_a\vdash\mathsf{m}_k}{C_a\vdash^+\mathsf{n}}\ (\mathsf{n}\leftarrow\mathsf{m}_1,\ldots,\mathsf{m}_k)\quad(\vdash^+\mathbf{C})$$

(b) Inference rules for message derivability

Fig. 3. Derivability: model of private/public keys, asymmetric encryption and lists (top); inference rules for message derivability (bottom) (m, n, m_i, k^- context messages; r_i bitstrings).

4.3 Profiles of Personal Information: Detectability and Linkability

Detectability and linkability capture the sets of personal information that an actor can compile about a data subject. Detectability $C_a \vdash \mathsf{p}$ means that actor a can determine the piece of information $p \in \mathsf{P}^c$. Linkability $*|^{\pi}_k \leftrightarrow_a *|^{\kappa}_l$ means the actor can conclude that the contexts $*|^{\pi}_k$, $*|^{\kappa}_l$ refer to the same person. An actor compiles information about a data subject by taking detectable context items whose contexts are mutually linkable.

Defining detectability and linkability using just derivability is not enough: actors may learn additional information and links from observing that items have the same contents. By the structural equivalence assumption, content equivalent messages can only be obtained by construction from content equivalent submessages. For instance, $E_{\mathsf{n}}(\mathsf{p}_1) \doteq E_{\mathsf{n}}(\mathsf{p}_2)$, implies that $\mathsf{p}_1 \doteq \mathsf{p}_2$. Thus, an actor who observes the two encryptions, learns information about the plaintexts. Formally:

Definition 4. *Let* $\mathsf{m}_1, \mathsf{m}_2$ *be two context messages, and* $\mathsf{p}_1, \mathsf{p}_2$ *two context items.*

- *We write* $(\mathsf{m}_1 \doteq \mathsf{m}_2) \Rightarrow (\mathsf{p}_1 \doteq \mathsf{p}_2)$ *if* $\mathsf{m}_1 \doteq \mathsf{m}_2$*, and* p_1 *and* p_2 *occur in the same location of* m_1 *and* m_2*, respectively (and hence,* $\mathsf{p}_1 \doteq \mathsf{p}_2$*).*
- *Let* C_a *be the knowledge base of actor a.* p_1 *and* p_2 *are* directly known to be content equivalent *by a, denoted* $\mathsf{p}_1 \doteq_a \mathsf{p}_2$*, if* $C_a \vdash \mathsf{m}_1$*,* $C_a \vdash \mathsf{m}_2$*, and* $(\mathsf{m}_1 \doteq \mathsf{m}_2) \Rightarrow (\mathsf{p}_1 \doteq \mathsf{p}_2)$.

(The fact that m_1,m_2 need to have derivations of the form of Definition 3 is not a restriction: also allowing constructed messages would give the same $\doteq_a$ relation.)

If an actor knows that two identifiers from different contexts are content equivalent, then he can conclude that the two contexts refer to the same data subject:

Definition 5. Direct linkability $\leftrightarrow_a$ *is the smallest equivalence relation on contexts s.t.:*

- *If* $\mathbf{id}|^{\pi}_{a_1,\ldots,a_k;\emptyset} \doteq_a \mathbf{id}|^{\kappa}_{b_1,\ldots,b_k;\emptyset}$ *or* $id|^{\pi}_{a_1,\ldots,a_k;\emptyset} \doteq_a id|^{\kappa}_{b_1,\ldots,b_k;\emptyset}$*, then* $\forall i$: $*|^{\pi}_{a_i} \leftrightarrow_a *|^{\kappa}_{b_i}$;
- *If* $id|^{\pi}_{a_1,\ldots,a_k;c_1,\ldots,c_m} \doteq_a id|^{\kappa}_{b_1,\ldots,b_k;d_1,\ldots d_m}$ *and* $\forall i$: $*|^{\pi}_{c_i} \leftrightarrow *|^{\kappa}_{d_i}$*, then* $\forall i$: $*|^{\pi}_{a_i} \leftrightarrow_a *|^{\kappa}_{b_i}$.

Detectability follows from $\vdash\!\cdots$ and $\doteq_a$; linkability follows from $\leftrightarrow_a$:

Definition 6. *Let C_a be a set of context messages; p a context item; $*|^{\pi}_{k}, *|^{\kappa}_{l}$ two contexts.*

- *We write $C_a \vdash \mathsf{p}$ (*p *is* detectable*) if $C_a \vdash\!\cdots \mathsf{p}$, or if $\mathsf{p} \doteq_a \mathsf{p}'$ for some p' with $C_a \vdash \mathsf{p}'$.*
- *We write $*|^{\pi}_{k} \leftrightarrow_a *|^{\kappa}_{l}$ ($*|^{\pi}_{k}$, $*|^{\kappa}_{l}$ are* linkable*) if $*|^{\pi}_{k} \leftrightarrow_a *|^{\kappa}_{l}$; or if there exists context context $*|^{\nu}_{m}$ such that $*|^{\pi}_{k} \leftrightarrow_a *|^{\nu}_{m}$ and $*|^{\nu}_{m} \leftrightarrow_a *|^{\kappa}_{l}$.*

(Note that $\vdash\!\cdots$ and $\leftrightarrow_a$ are defined inductively; hence any finite number of $\doteq_a$ and $\leftrightarrow_a$ steps may be made.)

Example 3. Consider an actor with knowledge base $C_a = \{\mathsf{p}_1, E_{\mathsf{pk(k^-)}}(\mathsf{p}_1), E_{\mathsf{pk(k^-)}}(\mathsf{p}_2)\}$, where the two encryptions are content equivalent. Detectability $C_a \vdash \mathsf{p}_1$ holds directly. Because the actor can conclude content equivalence $\mathsf{p}_1 \doteq_a \mathsf{p}_2$ from the two encryptions, also $C_a \vdash \mathsf{p}_2$. Moreover, suppose that $\mathsf{p}_1 = i|^{\pi}_{k;\emptyset}$, $\mathsf{p}_2 = i|^{\kappa}_{l;\emptyset}$ are identifiers: then $*|^{\pi}_{k} \leftrightarrow *|^{\kappa}_{l}$. Note that linkability can be concluded also if $\mathsf{p}_1 \notin C_a$. □

5 Reasoning using the Symbolic Protocol Model

In this section, we derive privacy properties for symbolic protocols that apply to any instantiation. Symbolic derivability expresses what messages can be derived from a protocol instance under what constraints (§5.1). Analogously, constraints can be defined for detectability and linkability (§5.2). Constraint graphs visualise all constraints relevant for detectability and linkability in a set of symbolic protocols (§5.3).

5.1 Symbolic Derivability

Symbolic derivability captures what symbolic messages can be derived from a protocol instance under what conditions. Conditions are modelled by *derivation constraints*: boolean formulae with conjunction $\wedge$, disjunction $\vee$ and two types of atomic propositions: external and internal constraints. *External constraint* $\mathfrak{m}$ expresses that contents of message $\mathfrak{m}$ need to be known apart from the protocol instance. *Internal constraint* $\mathfrak{m} \doteq^{i} \mathfrak{m}'$, expresses that messages $\mathfrak{m}, \mathfrak{m}'$ must have the same contents inside the protocol instance. **T** and **F** denote true and false.

Definition 7. *Let γ be a derivation constraint, and (C_a, π) an instantiation of $\mathfrak{C}$. Then γ is* satisfied *in (C_a, π) if: (i) $\gamma = \boldsymbol{T}$; (ii) $\gamma = \mathfrak{m}$, and $C_a \vdash\!\cdots \mathsf{m}'$ for some m' outside of domain π with $\mathfrak{m}|^{\pi} \doteq \mathsf{m}'$; (iii) $\gamma = \mathfrak{m} \doteq^{i} \mathfrak{m}'$ and $\mathfrak{m}|^{\pi} \doteq \mathfrak{m}'|^{\pi}$; (iv) $\gamma = \gamma_1 \vee \gamma_2$ and γ_1 or γ_2 is satisfied; or (v) $\gamma = \gamma_1 \wedge \gamma_2$ and γ_1 and γ_2 are satisfied.*

Definition 8. *Let $\mathfrak{C}$ be a set of symbolic messages, $\mathfrak{m}$ a symbolic message; and γ a derivation constraint. Then: (i) ϕ is* sufficient *for $\mathfrak{m}$ if whenever γ is satisfied in instantiation (C_a, π) of $\mathfrak{C}$, then $C_a \vdash\!\cdots \mathfrak{m}|^{\pi}$. (ii) ϕ is* necessary *for $\mathfrak{m}$ if whenever $C_a \vdash\!\cdots \mathfrak{m}|^{\pi}$ in instantiation (C_a, π) of $\mathfrak{C}$, then γ is satisfied.*

Constraints are treated as boolean formulae; *trivial* constraints (i.e., constraints satisfied in any instantiated model) and *non-satisfiable* constraints (i.e., constraints satisfied in no instantiated model) can be replaced by **T** and **F**. We find sufficient derivation constraints by means of a deductive system similar to the one for derivability in the instantiated model. The disjunction of these constraints is both necessary and sufficient.

$$\frac{}{\mathfrak{C}\vdash \mathbf{T}\Rightarrow \mathfrak{m}}\ (\mathfrak{m}\in\mathfrak{C})\ (\vdash\mathbf{0}) \qquad \frac{\mathfrak{C}\vdash \mathsf{req}_1\Rightarrow\mathfrak{m}}{\mathfrak{C}\vdash \mathsf{req}_1\wedge\mathsf{req}_2\Rightarrow\mathfrak{n}}\ (\mathfrak{m}\xrightarrow{\mathsf{req}_2}\mathfrak{n})\ (\vdash\mathbf{E}) \qquad \frac{\mathfrak{C}\vdash \mathsf{req}\Rightarrow\{\mathfrak{m}\}_F}{\mathfrak{C}\vdash \mathsf{req}\Rightarrow\{\mathfrak{m}\}_F@k}\ (\vdash@)$$

$$\frac{\mathfrak{C}\vdash \mathfrak{r}_1\wedge\ldots\wedge\mathfrak{r}_k\Rightarrow\mathfrak{m}\quad C\vdash \mathsf{req}\Rightarrow\mathfrak{r}'_i}{\mathfrak{C}\vdash(\mathfrak{r}_1\wedge\ldots\wedge\mathfrak{r}_{i-1}\wedge\mathfrak{r}_{i+1}\wedge\ldots\wedge\mathfrak{r}_k\wedge\mathsf{req}\wedge\mathfrak{r}_i\doteq^i\mathfrak{r}'_i)\Rightarrow\mathfrak{m}}\ (\vdash\mathbf{RE})$$

$$\frac{\mathfrak{C}\vdash \mathfrak{r}_1\wedge\ldots\wedge\mathfrak{r}_k\Rightarrow\mathfrak{m}}{\mathfrak{C}\vdash \mathfrak{r}_1\wedge\ldots\wedge\mathfrak{r}_{i-1}\wedge\mathfrak{r}_{i+1}\wedge\ldots\wedge\mathfrak{r}_k\wedge\mathfrak{r}'_1\wedge\ldots\wedge\mathfrak{r}'_l\Rightarrow\mathfrak{m}}\ (\mathfrak{r}_i\leftarrow\mathfrak{r}'_1,\ldots,\mathfrak{r}'_l)\quad(\vdash\mathbf{RC})$$

Fig. 4. Inference rules for the symbolic derivability relation ($\mathfrak{C}$ a set of symbolic messages; req, req_i sets of derivation constraints; $\mathfrak{m}$, $\mathfrak{m}_i$, $\mathfrak{r}_i$, $\mathfrak{r}'_i$, $\mathfrak{n}$ symbolic messages, k any index)

Definition 9. *Let* $\mathfrak{C}$ *be a set of symbolic messages,* $\mathfrak{m}$ *a symbolic message, and* req *a derivation constraint. We say that* $\mathfrak{m}$ *is* symbolically derivable *from* $\mathfrak{C}$ *using* req*, denoted* $\mathfrak{C}\vdash\mathsf{req}\Rightarrow\mathfrak{m}$*, if the conclusion* $\mathfrak{C}\vdash\mathsf{req}\Rightarrow\mathfrak{m}$ *follows from the deductive system of Figure 4.*

The first two inference rules of Figure 4 mimic those for context messages (Figure 3(b)); the other three are specific to symbolic derivability. Rule ($\vdash\mathbf{0}$) is the direct analogue of its instantiated counterpart. Rule ($\vdash\mathbf{E}$) represents application of a cryptographic operation by adding its requirements to the constraints. Rule ($\vdash@$) is a special "elimination rule" for deriving an element of a variable-length list. The rules ($\vdash\mathbf{RE}$) and ($\vdash\mathbf{RC}$) are new symbolic rules to replace external constraints by internal ones. Namely, external constraint $\mathfrak{r}_i$ can be replaced by internal constraint $\mathfrak{r}_i\doteq^i\mathfrak{r}'_i$ and the derivation constraints for $\mathfrak{r}'_i$ ($\vdash\mathbf{RE}$). Also, external constraint $\mathfrak{m}$ can be replaced by the messages it is constructed from ($\vdash\mathbf{RC}$); ($\vdash\mathbf{RE}$) can then be applied to these messages. (The $\rightarrow$ and $\leftarrow$ relations are interpreted as relations on symbolic messages in the obvious way.)

Proposition 1. *Let* $\mathfrak{C}$ *be a set of symbolic messages, and* $\mathfrak{m}$ *a symbolic message. If* $\mathfrak{C}\vdash\mathsf{req}\Rightarrow\mathfrak{m}$*, then* req *is sufficient for* $\mathfrak{m}$*. Let* $\mathsf{req}_1,...,\mathsf{req}_k$ *be all* req *such that* $\mathfrak{C}\vdash\mathsf{req}\Rightarrow\mathfrak{m}$ *holds, and* $\mathfrak{C}\vdash\mathsf{req}\Rightarrow\mathfrak{m}$ *does not hold for any* $\mathsf{req}'\neq\mathsf{req}$ *implied by* req*. Then* $\mathsf{req}_1\vee...\vee\mathsf{req}_k$ *is both necessary and sufficient for* $\mathfrak{m}$*.*

Example 4. Consider the set $\mathfrak{C}=\{E_{\mathsf{pk}(\mathbf{k}^-|_{srv;\emptyset})}(d_1|_{u;\top}),\mathsf{pk}(\mathbf{k}^-|_{srv;\emptyset}),d_2|_{u;\top}\}$. The constraint $d_1|_{u;\top}\doteq d_2|_{u;\top}\vee d_1|_{u;\top}\vee\mathbf{k}^-|_{srv;\emptyset}$ is necessary and sufficient for $d_1|_{u;\top}$. For instance, symbolic derivability of $d_1|_{u;\top}$ using $d_1|_{u;\top}\doteq d_2|_{u;\top}$ is derived as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathfrak{C}\vdash\mathbf{T}\Rightarrow E_{\mathsf{pk}(\mathbf{k}^-|_{srv;\emptyset})}(d_1|_{u;\top})}(\vdash\mathbf{0_1})}{\mathfrak{C}\vdash d_1|_{u;\top}\wedge\mathsf{pk}(\mathbf{k}^-|_{srv;\emptyset})\Rightarrow d_1|_{u;\top}}(\vdash\mathbf{E}) \quad \dfrac{}{\mathfrak{C}\vdash\mathbf{T}\Rightarrow d_2|_{u;\top}}(\vdash\mathbf{0_2})}{\mathfrak{C}\vdash d_1|_{u;\top}\doteq^i d_2|_{u;\top}\wedge\mathsf{pk}(\mathbf{k}^-|_{srv;\emptyset})\Rightarrow d_1|_{u;\top}}(\vdash\mathbf{RE_1}) \quad \dfrac{}{\mathfrak{C}\vdash\mathbf{T}\Rightarrow\mathsf{pk}(\mathbf{k}^-|_{srv;\emptyset})}(\vdash\mathbf{0_3})}{\mathfrak{C}\vdash d_1|_{u;\top}\doteq^i d_2|_{u;\top}\Rightarrow d_1|_{u;\top}}(\vdash\mathbf{RE_2})$$

An actor with $\mathfrak{C}$ knows $E_{\mathsf{pk}(\mathbf{k}^-|_{srv;\emptyset})}(d_1|_{u;\top})$ ($\vdash\mathbf{0_1}$); from contents of $d_1|_{u;\top}$ and $\mathsf{pk}(\mathbf{k}^-|_{srv;\emptyset})$ he can derive $d_1|_{u;\top}$ ($\vdash\mathbf{E}$). He also knows $d_2|_{u;\top}$ ($\vdash\mathbf{0_2}$), so if $d_1|_{u;\top}\doteq^i d_2|_{u;\top}$, he can use $d_2|_{u;\top}$ as contents of $d_1|_{u;\top}$ ($\vdash\mathbf{RE_1}$). Finally, because he knows $\mathsf{pk}(\mathbf{k}^-|_{srv;\emptyset})$ ($\vdash\mathbf{0_3}$), constraint $\mathsf{pk}(\mathbf{k}^-|_{srv;\emptyset})$ is always true and can be eliminated ($\vdash\mathbf{RE_2}$). □

Symbolic derivability can be computed automatically using a Prolog tool. The tool starts with set $\mathfrak{C}$, and iteratively derives new symbolically derivable messages and additional constraints until there is no improvement. In practice, this computation is feasible because a) message sets $\mathfrak{C}$ are relatively small; b) finitely many construction and elimination rules apply to any given message; c) there are few satisfiable, sufficient constraints; and d) we only consider messages eliminated from $\mathfrak{C}$ (although deriving them may need construction). The tool is available at http://www.mobiman.me/downloads/.

5.2 Symbolic Knowledge of Content Equivalence and Direct Links

We now extend the above approach to knowledge of detectability and linkability. Detectability and linkability depend on knowledge of content equivalence of items from two contexts. For an actor to conclude content equivalence (Definition 4), he needs to derive a message in the first context; it needs to be content equivalent to a message in the second context; and he needs to derive the second message. Linkability additionally imposes that some actors in the two protocol instances are the same (Definition 5). We capture these different conditions in *content equivalence constraints*:

Definition 10. *A* content equivalence constraint *is a disjunction of conjunctions* $\gamma_1 \wedge \mathfrak{m} \doteq^c \mathfrak{m}' \wedge p_1 =^c p'_1 \wedge \ldots \wedge p_k =^c p'_k \wedge \gamma_2$*, where* γ_1*,* γ_2 *are derivation constraints;* $\mathfrak{m}$*,* $\mathfrak{m}'$ *are symbolic messages; and* p_i*,* p'_i *are profiles.* $\boldsymbol{T}$ *denotes true and* $\boldsymbol{F}$ *denotes false.*

Definition 11. *Let* $\mathfrak{C}, \mathfrak{C}'$ *be sets of symbolic messages;* $\mathfrak{p}, \mathfrak{p}'$ *symbolic items; and* γ *a content equivalence constraint. Let* (C_a, π) *and* (C_a, κ) *be instantiations of* $\mathfrak{C}$ *and* $\mathfrak{C}'$*, respectively. Then* γ *is* satisfied *in* (C_a, π)*,* (C_a, κ) *if: (i)* $\gamma = \boldsymbol{T}$*; (ii)* $\gamma = (\gamma_1 \wedge \mathfrak{m} \doteq^c \mathfrak{m}' \wedge p_1 =^c p'_1 \wedge \ldots \wedge p_k =^c p'_k \wedge \gamma_2)$*, and:* γ_1 *is satisfied in* (C_a, π)*,* $\mathfrak{m}|^\pi \doteq \mathfrak{m}'|^\kappa$*,* $*|^\pi_{p_i} \leftrightarrow *|^\kappa_{p'_i}$ *for all i, and* γ_2 *is satisfied in* (C_a, κ)*; or (iii)* $\gamma = \gamma_1 \vee \gamma_2$ *and* γ_1 *or* γ_2 *is satisfied.*

Definition 12. *Let* $\mathfrak{C}, \mathfrak{C}'$ *be sets of symbolic messages;* $\mathfrak{p}, \mathfrak{p}'$ *symbolic items, and* γ *a content equivalence constraint. (i)* γ *is* sufficient *for* $\mathfrak{p} \doteq_a \mathfrak{p}'$ *if whenever* γ *is satisfied in instantiations* (C_a, π)*,* (C_a, κ) *of* $\mathfrak{C}$*,* $\mathfrak{C}'$*, then* $\mathfrak{p}|^\pi \doteq_a \mathfrak{p}'|^\kappa$*; (ii)* γ *is* necessary *for* $\mathfrak{p} \doteq_a \mathfrak{p}'$ *if whenever* $\mathfrak{p}|^\pi \doteq_a \mathfrak{p}'|^\kappa$ *in instantiations* (C_a, π)*,* (C_a, κ) *of* $\mathfrak{C}$*,* $\mathfrak{C}'$*, then* γ *is satisfied.*

Conditions $p_i =^c p'_i$ can be omitted in sufficient constraints for content equivalence; however, adding the constraints arising from content equivalence of random global identifiers clarifies which actors need to be involved in the protocol. Because variable-length lists can be empty, constraints (other than **F**) sufficient for content equivalence of variable-length list items may not exist. Again, we consider only satisfiable constraints.

Given symbolic items $\mathfrak{p}$ in profile $\mathfrak{P}(r)|_p$ and $\mathfrak{p}'$ in profile $\mathfrak{P}'(r')|_{p'}$, we can find a necessary content equivalence constraint for $\mathfrak{p} \doteq_a \mathfrak{p}'$ as follows. Find all $\mathfrak{m}_i$ derivable from $\mathfrak{P}(r)|_p$ that contain $\mathfrak{p}$, and their necessary and sufficient derivability constraints req_i. Similarly for req'_j, $\mathfrak{m}'_j$. Consider set I of pairs (i, j) such that there exists an instantiated model in which $(\mathfrak{m}_i|^\pi \doteq \mathfrak{m}'_j|^\kappa) \Rightarrow (d|^\pi_{P;\top} \doteq d'|^\kappa_{P;\top})$. Then $\vee_{(i,j)\in I}(\mathsf{req}_i \wedge \mathfrak{m}_i \doteq^c \mathfrak{m}'_j \wedge \mathsf{req}'_j)$ is a necessary content equivalence constraint for $d|_{P;\top} \doteq_a d'|_{P';\top}$. As noted above, $p_i =^c p'_i$-type constraints can be added after each $\mathfrak{m}_i \doteq^c \mathfrak{m}'_j$ to clarify actors' involvement.

Definition 13. *Let* $\mathfrak{C}, \mathfrak{C}'$ *be sets of symbolic messages,* $*|_p, *|_q$ *profiles, and* γ *a content equivalence constraint. (i)* γ *is* sufficient *for* $*|_p \leftrightarrow_a *|_q$ *if, whenever* γ *is satisfied in*

instantiations (C_a, π), (C_a, κ) *of* $\mathfrak{C}$, $\mathfrak{C}'$, *then* $*|^{\pi}_{p} \leftrightarrow_a *|^{\kappa}_{q}$*; (ii)* γ *is* necessary *for* $*|_p \leftrightarrow_a *|_q$ *if, whenever* $*|^{\pi}_{p} \leftrightarrow_a *|^{\kappa}_{q}$ *in instantiations* (C_a, π), (C_a, κ) *of* $\mathfrak{C}$, $\mathfrak{C}'$, *then* γ *is satisfied.*

Conditions $p_i{=}^c p'_i$ are needed in sufficient constraints for linkability using local identifiers. Similarly to above, we also include additional $p_i{=}^c p'_i$-type constraints arising from random global identifiers to clarify the involvement of different actors.

Necessary content equivalence constraints for $*|_p \leftrightarrow_a *|_q$ are found as follows. As above, find all necessary content equivalence constraints for $\mathfrak{i} \doteq_a \mathfrak{i}'$, with $\mathfrak{i}, \mathfrak{i}'$ identifiers. Suppose content equivalence constraint $\gamma = (\gamma_1 \wedge \mathfrak{m}{\doteq}^c\mathfrak{m}' \wedge \gamma_2)$ is used to conclude $\mathfrak{i} \doteq_a \mathfrak{i}'$. If identifiers $\mathfrak{i}, \mathfrak{i}'$ are local, i.e., $\mathfrak{i} = \{id|_{a_1,\ldots,a_k;b_1,\ldots,b_l}\}$, $\mathfrak{i}' = id|_{a'_1,\ldots,a'_k;b'_1,\ldots,b'_l}$, then let $\gamma' = \gamma_1 \wedge \mathfrak{m}{\doteq}^c\mathfrak{m}' \wedge b_1{=}^c b'_1 \wedge \ldots \wedge b_l{=}^c b'_l \wedge \gamma_2$; otherwise, let $\gamma' = \gamma$. The disjunction of all γ' from all identifier pairs is a necessary content equivalence constraint for $*|_p \leftrightarrow_a *|_q$.

Our Prolog tool helps to determine content equivalence constraints. Namely, it indicates possibly content equivalent messages, as well as $p_i{=}^c p'_i$-type conditions needed for concluding linkability. From this information, constraints are easily determined.

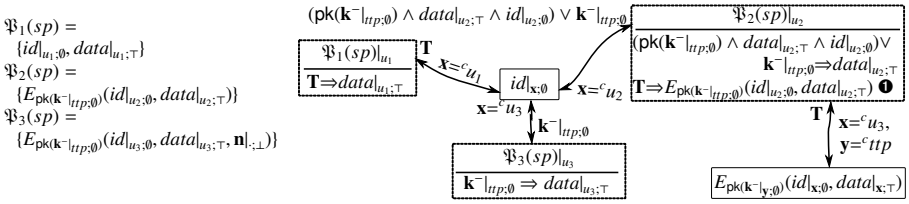
5.3 Symbolic Analysis of Detectability and Linkability: The Constraints Graph

Constraints relevant for detectability and linkability in an information system (i.e. a set of symbolic protocol roles) are shown in its *constraints graph*. The graph has *profile nodes* $\mathfrak{P}(r)|_p$ for each profile p in protocol role $\mathfrak{P}(r)$; *id-nodes* for messages used to link different profiles; and edges connecting these messages to profiles they occur in.

Profile nodes $\mathfrak{P}(r)|_p$ summarise the constraints relevant for detectability of data items. Data item $\mathfrak{d}$ can be detected if it can be derived, or if it can be concluded content equivalent to another detectable data item. For derivability of data item $\mathfrak{d}$, the profile node shows its derivation constraints. For content equivalence conclusions, it shows all messages $\mathfrak{m}$ that contain $\mathfrak{d}$ and can occur outside of the protocol instance. (Or: that can be used to conclude content equivalence of $\mathfrak{d}$ to another item *within* the protocol instance.) These messages are numbered ❶, ❷, ... Messages $\mathfrak{m}, \mathfrak{m}'$ with the same number may be content equivalent and hence useful for concluding content equivalence; a necessary content equivalence constraint for $\mathfrak{d} \doteq_a \mathfrak{d}'$ is thus obtained by collecting the constraints from all equally-numbered pairs $(\mathfrak{m}, \mathfrak{m}')$ that contain $(\mathfrak{d}, \mathfrak{d}')$.

Decision Procedure 1 *Data item* $\mathfrak{d}$ *is detectable only if one of the following three conditions holds: (i)* $\mathfrak{d}$*'s derivation constraints shown in the graph are satisfied; or (ii)* $\mathfrak{d}$ *is directly known to be content equivalent to some* $\mathfrak{d}'$ *using same-numbered messages* $\mathfrak{m}, \mathfrak{m}'$ *shown in the graph with satisfied constraints, and* $\mathfrak{d}'$ *is detectable; or (iii) derivation constraints for message* $\mathfrak{m}$ *containing* $\mathfrak{d}$ *shown in the graph are satisfied, and* $\mathfrak{m}$ *re-occurs along with* $\mathfrak{d}$ *outside of the system.*

Edges from two profile nodes $\mathfrak{P}(r)|_p$, $\mathfrak{P}'(r')|_{p'}$ to one id-node n indicate possible direct linkability $*|_p \leftrightarrow_a *|_{p'}$. As shown above, the necessary content equivalence constraint for $*|_p \leftrightarrow_a *|_{p'}$ is a disjunction of conjuncts $\gamma_1 \wedge \mathfrak{m}{\doteq}^c\mathfrak{m}' \wedge p_1{=}^c p'_1 \wedge \ldots \wedge p_k{=}^c p'_k \wedge \gamma_2$. Each $\mathfrak{m}{\doteq}^c\mathfrak{m}'$ corresponds to an id-node n. The edge from $\mathfrak{P}(r)|_p$ to n is labelled by γ_1 and $\mathbf{x}{=}^c p_1, \mathbf{y}{=}^c p_2, \ldots$; the edge from $\mathfrak{P}'(r')|_{p'}$ to n is labelled by γ_2 and $\mathbf{x}{=}^c p'_1, \mathbf{y}{=}^c p'_2, \ldots$. Node n is labelled by a single representation of $\mathfrak{m}$ and $\mathfrak{m}'$ in which $p_1, p_2, \ldots$ are replaced by $\mathbf{x}, \mathbf{y}, \ldots$. (Usually, $\mathfrak{m}$ and $\mathfrak{m}'$ do not differ in other places; otherwise, other differences are indicated with additional variables.)



(a) Symbolic protocol roles (b) Constraints graph

Fig. 5. Analysis of three variants of a simple authentication protocol (Example 5)

Decision Procedure 2 *Profiles $\mathfrak{P}(r)|_p$, $\mathfrak{P}'(r')|_{p'}$ are linkable only if either (i) their profile nodes are connected via an id-node with satisfied constraints; or (ii) $\mathfrak{P}(r)|_p$ is connected to another profile node q via an id-node with satisfied constraints, and q is linkable to $\mathfrak{P}'(r')|_{p'}$; or (iii) $\mathfrak{P}(r)|_p$ is connected to an id-node whose message re-occurs outside of the information system, and a link to $\mathfrak{P}'(r')|_{p'}$ can be established from there.*

Example 5. Consider an information system with three "authentication protocols" $\mathfrak{P}_i$, $i = 1, 2, 3$. In each variant i, service provider sp receives identifier $id|_{u_i;\emptyset}$ and data $data|_{u_i;\top}$ about user u_i. In variant $\mathfrak{P}_1$, the identifier and data are sent directly. In variant $\mathfrak{P}_2$, the identifier and data are encrypted for a trusted third party ttp using its public key $\mathsf{pk}(\mathbf{k}^-|_{ttp;\emptyset})$. In variant $\mathfrak{P}_3$, a nonce $\mathbf{n}|_{\cdot;\perp}$ is added to ensure the encryption is different every time. We analyse the knowledge of a service provider who runs different instances of the three protocols. Figure 5 shows the model of the system and its constraints graph.

Detectability of $data|_{u_i;\top}$ is seen from the profile nodes. The data $data|_{u_2}$ in $\mathfrak{P}_2$ can be detected by either decrypting or reconstructing the encryption; it may also be detectable if encryption ❶ occurs in other instances of $\mathfrak{P}_2$ or outside of the system. In $\mathfrak{P}_3$, because of nonce $\mathbf{n}|_{\cdot;\perp}$, decryption using $\mathbf{k}^-|_{ttp;\emptyset}$ is the only possibility. Instances of $\mathfrak{P}_1$ with the same identifier are linkable. Instances of $\mathfrak{P}_3$ are linkable to each other, or to instances of other protocols, only if $\mathbf{k}^-|_{ttp;\emptyset}$ is known. Instances of $\mathfrak{P}_2$ with the same identifier, data and TTP are mutually linkable because of the deterministic encryption. □

In the above example, we only showed the sp role and the u profile of the protocols. Other roles and profiles may be added; in particular, different profiles in the same protocol (instance) may be linkable. To analyse coalitions of actors who perform multiple roles p_1, p_2 in a single instance of protocol $\mathfrak{P}$, include nodes $\mathfrak{P}(p_1) \cup \mathfrak{P}(p_2)|_u$. Constraints graphs are easily built from the constraints computed by our Prolog tool.

6 Case Study: Identity Mixer

In this section, we analyse privacy in Identity Mixer [1]. We present the system and relevant privacy properties (§6.1); our formal model (§6.2); and analysis results (§6.3).

6.1 Identity Mixer

Identity Mixer [1] is an identity management system in which all exchange of personal information is via the user. In an identity management system, service providers

(SPs) delegate the task of authenticating a user and endorsing attributes about him to an *identity provider* (IdP). Traditionally, SPs and IdPs communicate directly in every transaction, facilitating user profiling by IdPs [19]. In Identity Mixer, by contrast, IdPs periodically *issue* anonymous credentials to the user containing all her attributes; during a transaction, the user *shows* a selection of attributes contained in these anonymous credentials to the SP without involving the IdP. We model a particular variant of Identity Mixer with two types of IdP: one can revoke user anonymity with a Trusted Third Party (TTP) [4], one cannot. The user shows attributes from two IdPs: one of each type.

We verify whether Identity Mixer satisfies several privacy properties of identity management systems. These properties have been elicited in [19] through a study of privacy claims for Identity Mixer and other systems, and privacy taxonomies. We check that SPs only learn those attributes (*irrelevant attribute undetectability*) and those properties of attributes (*property-attribute undetectability*) that are selected by the user. We check that different user profiles cannot be linked: different credential showings (involving one or more SPs: *session unlinkability*); credential showings to credential issuings (*IdP-SP unlinkability*), and issuings by different identity providers (*IdP profile unlinkability*). We also check that SPs *do* learn the attributes and properties they should learn (*attribute exchange*), and that anonymity revocation *can* be performed (*anonymity revocation*). (Identity Mixer trivially satisfies the other properties mentioned in [19] because of the lack of direct contact between IdPs and SPs. We do not consider them here.)

6.2 Formal Model

We model Identity Mixer as issuing protocol variants $\mathfrak{Iss}$, $\mathfrak{Iss}'$ between user u and IdP idp, and showing protocol $\mathfrak{Show}$ between user u and SP sp. The showing protocol involves IdPs $idp1$, $idp2$ whose credentials are shown; and TTP ttp (Figure 6, top-left).

The user has an identifier $i_{rev}|_{u;idp}$ at each identity provider and a global identifier $\mathbf{i}|_{u;\emptyset}$ he generates himself. Variable-length lists $\{d_i|_{u;\top}\}_F$ with various families F contain the user's attributes. For issuing, we use $F = all$ (all attributes known to the IdP). For showing, we use $F = a_1$ (all attributes from the first IdP), $F = d_1$ (disclosed attributes from the first IdP), $F = pr_1$ (disclosed properties from the first IdP), $F = pd_1$ (attributes of which a property is disclosed from the first IdP), $F = nd_1$ (non-disclosed attributes from the first IdP), and similarly for $F = a_2$, ... Item $td|_{u;\top}$ represents transaction details.

In $\mathfrak{Iss}$, $\mathfrak{Iss}'$, the user generates a commitment $\mathrm{SK0}_{\mathsf{pk}(\mathbf{k}^-|_{idp;\emptyset})}(\mathbf{i}|_{u;\emptyset}; \mathbf{n_{c,2}}|_{u,idp;\emptyset})$ to her identifier using random data $\mathbf{n_{c,2}}|_{u,idp;\emptyset}$ and public key $\mathsf{pk}(\mathbf{k}^-|_{idp;\emptyset})$ of the IdP. From this commitment, the IdP generates signature $S_{\mathbf{k}^-|_{idp;\emptyset}}(\mathbf{i}|_{u;\emptyset}, \mathbf{z}, \{d|_{u;\top}\}_{all}, \mathbf{n_{c,2}}|_{u,idp;\emptyset}, \mathbf{n_{c,5}}|_{u,idp;\emptyset})$ with additional randomness $\mathbf{n_{c,5}}|_{u,idp;\emptyset}$. In $\mathfrak{Iss}$, $\mathbf{z} = i_{rev}|_{u;idp}$ (revocation possible); in $\mathfrak{Iss}'$, $\mathbf{z} = \{\}$ (revocation not possible). Randomness $\mathbf{n_{c,2}}|_{u,idp;\emptyset}$, $\mathbf{n_{c,5}}|_{u,idp;\emptyset}$ is unique for the combination u, idp and hence an identifier of both. In $\mathfrak{Show}$, anonymity revocation is achieved by encrypting $i_{rev}|_{u;idp1}$ using the TTP's public key $\mathsf{pk}(\mathbf{k}^-|_{ttp;\emptyset})$. Further details (omitted due to lack of space) are in a technical report [21].

6.3 Privacy Analysis using the Constraints Graph

Figure 6 shows the constraints graph for the user profile in $\mathfrak{Iss}(idp)$, $\mathfrak{Iss}'(idp)$, and $\mathfrak{Show}(sp)$. This is sufficient for privacy analysis: knowledge about roles other than the

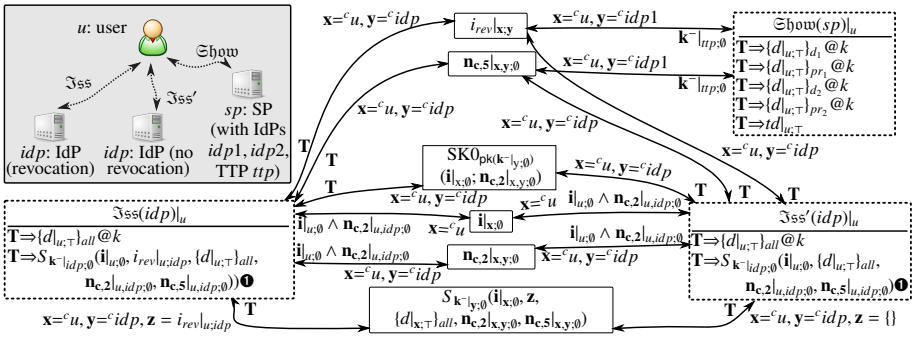


Fig. 6. Identity Mixer: roles and symbolic protocols (top-left) and constraints graph

user (profile nodes $\mathfrak{Iss}(idp)|_{idp}$, etc.) and knowledge of the user itself (profile nodes $\mathfrak{Iss}(u)|_u$, etc.) are not relevant. (In this system, user nodes $*(*)|_u$ and nodes $*(*)|_p$, $p \neq u$ are never directly linkable. In general, such a possibility does need to be considered.)

The constraints graph shows the following privacy guarantees for Identity Mixer. *Irrelevant attribute undetectability* and *property-attribute detectability* hold: there are no satisfiable derivation constraints for $\{d_i|_{u;\top}\}_F @k$ for $F \in \{a1, pd1, nd1, a2, pd2, nd2\}$ in $\mathfrak{Show}(sp)$ (so no derivability), nor for messages containing them (so no detectability by content equivalence). *Session unlinkability*, i.e., the impossibility to link user profiles in different instances of $\mathfrak{Show}(sp)$, holds if the TTP's secret key is kept secret (or if the identity providers are different). *IdP-SP unlinkabiltity* also holds if the TTP's secret key is kept secret; in any case the IdP needs revocation capability. *IdP profile unlinkability* can only be broken inside the system via a service provision (but this requires the TTP's secret key) or using $\mathbf{i}|_{\mathbf{x};\emptyset}$ (but this requires random data that the user should keep secret). It can also be broken via knowledge outside the system; because the user authenticates to the IdP to get his credential, this is unavoidable. Positive linkability and detectability results can also be established: $\{d|_{u;\top}\}_{d_i} @k$ and $\{d|_{u;\top}\}_{pr_i} @k$ $(i = 1, 2)$ are derivable with constraint $\mathbf{T}$, giving *attribute exchange*; $\mathfrak{Iss}(idp)$ and $\mathfrak{Show}(sp)$ are linkable by $i_{rev}|_{u;idp}, i_{rev}|_{u;idp1}$ using the TTP's secret key, giving *anonymity revocation*. In conclusion, Identity Mixer satisfies data minimisation under reasonable conditions.

7 Conclusions and Future Work

In this work, we presented a symbolic model for privacy analysis of communication protocols. In our previous work, we define an instantiated model of actor knowledge in which detectability and linkability properties are defined independently from any application; and we verified such properties in particular scenarios. In this paper, we build on this by reasoning symbolically about *any* scenario that instantiates a set of protocols subject to a set of basic, well-defined conditions. We keep the main advantages of the instantiated model: general privacy definitions that are verified in a single model. In addition, conclusions now apply to any possible scenario; unlike in other approaches, our model remains feasibly evaluatable even in this case. We implemented the model;

showed how its results can be visualised; and demonstrated feasibility by applying it to the Identity Mixer identity management system.

Acknowledgements This work is funded by the STW Mobiman project (#10522).

References

1. Bangerter, E., Camenisch, J., Lysyanskaya, A.: A Cryptographic Framework for the Controlled Release of Certified Data. In: Proceedings of SPW '04, Springer (2004)
2. Blanchet, B., Abadi, M., Fournet, C.: Automated Verification of Selected Equivalences for Security Protocols. J. Log. Algebr. Program. **75**(1) (2008) 3–51
3. Butin, D., Bella, G.: Verifying Privacy by Little Interaction and No Process Equivalence. In: Proceedings of SECRYPT '12, INSTICC Press (in press)
4. Camenisch, J., Sommer, D., Zimmermann, R.: A General Certification Framework with Applications to Privacy-Enhancing Certificate Infrastructures. In: Proceedings of SEC '06, Springer (2006)
5. Clarke, E.M., Jha, S., Marrero, W.R.: Using State Space Exploration and a Natural Deduction Style MessageDerivation Engine to Verify Security Protocols. In: Proceedings of PROCOMET '98, Chapman & Hall, Ltd. (1998)
6. Dahl, M., Delaune, S., Steel, G.: Formal Analysis of Privacy for Anonymous Location Based Services. In: Proceedings of TOSCA'11, Springer (2011)
7. Dong, N., Jonker, H., Pang, J.: Formal Analysis of Privacy in an eHealth Protocol. In: Proceedings of ESORICS '12, Springer (2012)
8. Dreier, J., Lafourcade, P., Lakhnech, Y.: A Formal Taxonomy of Privacy in Voting Protocols. Tech report, Verimag (2011)
9. Hansen, M., Berlich, P., Camenisch, J., Clauß, S., Pfitzmann, A., Waidner, M.: Privacy-Enhancing Identity Management. Inf. Secur. Tech. Rep. **9**(1) (2004) 35–44
10. Hoepman, J.H., Joosten, R., Siljee, J.: Comparing Identity Management Frameworks in a Business Context. In: Proceedings of IFIP/FIDIS summer school '08, Springer (2008)
11. Identity Management Systems (IMS): Identification and Comparison Study. Independent Centre for Privacy Protection Schleswig-Holstein (2003)
12. Meadows, C.: Formal Methods for Cryptographic Protocol Analysis: Emerging Issuesand Trends. IEEE Sel. Areas Commun. **21**(1) (2003) 44–54
13. Data protection guidelines on research in the health sector. Office of the Data Protection Commissioner of Ireland (2007)
14. Paulson, L.C.: The Inductive Approach to Verifying Cryptographic Protocols. Comput. Secur. **6**(1-2) (1998) 85–128
15. 2011 Cost of Data Breach Study: Global. Ponemon Institute (2011)
16. 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 ICITST '09, IEEE (2009)
17. Prescription drug data: HHS has issued health privacy and security regulations but needs to improve guidance and oversight. U.S. Govt. Accountability Office (2012)
18. Veeningen, M., de Weger, B., Zannone, N.: Formal Modelling of (De)Pseudonymisation: A Case Study in Health Care Privacy. In: Proceedings of STM '12. Springer (2012)
19. Veeningen, M., de Weger, B., Zannone, N.: A Formal Privacy Analysis of Identity Management Systems. Tech report, ArXiv.org (2012)
20. Veeningen, M., Zannone, N., de Weger, B.: Formal Privacy Analysis of Communication Protocols for Identity Management. In: Proceedings of ICISS '11. LNCS 7093, Springer (2011) 235–249
21. Veeningen, M.: Symbolic Analysis of Identity Mixer. Tech report, www.mobiman.me (2013)