Formal Policy-based Provenance Audit

Denis Butin, Denise Demirel, and Johannes Buchmann

TU Darmstadt, Germany

TECHNISCHE
UNIVERSITÄT
DARMSTADT

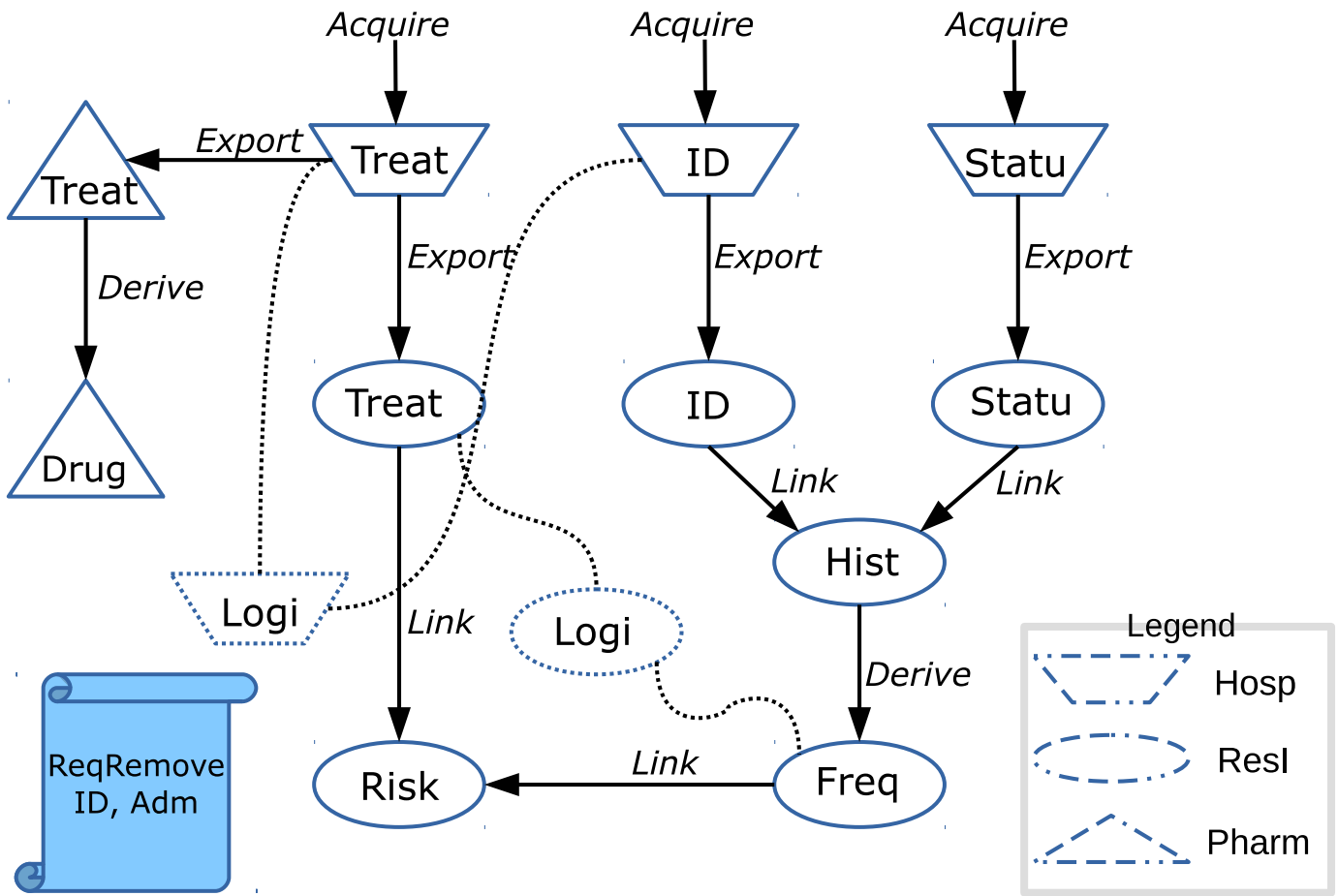
Context (1)

- Data *provenance*: often defined as origin & history of data, or as process-related metadata
- We adopt W3C definition: *(. . .) a record that describes the people, institutions, entities and activities, involved in producing, influencing, or delivering a piece of data or a thing*
- Key idea: comprehensive picture of data life cycle, from creation to destruction
- Includes data usage, forwarding, and the creation of new data on the basis of existing information

Context (2)

- Provenance is valuable to verify compliance with usage policies (rules) for sensitive data processed by several related entities
- In such large networks, data flow easily becomes opaque, making simple audit methods unadapted
- Provenance can help demonstrate compliance with policies (*accountability*)
- Allows auditor to detect system failures
- We discusser later and separately the case of personal data — usage policies are then called *privacy policies*

A Simplified Provenance Graph (3 Components)



Different shapes symbolise different components; dotted lines represent Use events.

Motivations

- Provenance as a tool to audit compliance with regulations (e.g. privacy regulations in the case of personal data)
- Formal approaches lacking so far
- Consistency of provenance records and their compliance with associated usage policies must be stated precisely as a first step toward semi-automated analysis

In this work (1):

- Formal framework modelling provenance events and their compliance with usage policies
- First introduce usage policies, modelled as tuples of constraints and permissions
- Then model provenance records as sequences of discrete events (single data subject, but multiple entities and data categories)

In this work (2):

- Formalise correctness and compliance of logs of provenance events with respect to usage policies
- Correctness: internal consistency, independently of policies
- Compliance: relation between provenance record and associated policies
- Finally, we discuss limitations for privacy case and integration in global accountability process

Definitions

- *Component*: an entity collecting or processing data
- *Usage policy*: a machine-readable set of rules regarding the use of data
- *Data category*: designation of the type of data in natural language, e.g. `postal address`
- *Purpose*: finality or goal of an operation

Usage Policies (1)

- For systematic evaluation of provenance records, formalisation of usage policies introduced
- Usual policy languages such as PPL do not support various data handling operations required when considering provenance
- *Derivation*: transformation of one entity into another
- *Linking*: combination of different categories of data from a single source
- *Forwarding*: sending of data from one component to another (sending component also retains data)

Usage Policies (2)

Provenance usage policies are defined as tuples:

$$\mathcal{P} = \mathsf{D_g} \times \mathsf{D_{rf}} \times \mathsf{Fw} \times \mathsf{Li} \times \mathsf{De} \times \mathsf{P_{use}} \times \mathsf{P_{der}}$$

- $\mathsf{D_g}$: global deletion delay — maximal delay for deletion of data of any category
- $\mathsf{D_{rf}}$: request fulfilment delay (e.g. for data deletion requests)
- $\mathsf{Fw} = (\text{restriction}, \text{List})$: forwarding policy. Possible values for restriction: $\bot$ (data may be fwd to any component), $\top$ (data may be fwd to no other component), bl (list of forbidden components declared) and wl (list of authorised components declared). Where applicable, List names components *to* which data fwd is forbidden or permitted, resp.

Usage Policies (3)

- Li: linking policy — set of pairs limting linking of data categories by components. If two data categories are in same pair of this set, pieces of data with these categories may never be linked, directly or indirectly
- De: derivation policy — set of data categories. Pieces of data with categories in set may never be used to derive new data, directly or indirectly
- P_{use}: authorised use purposes — set of pairs (data category, acceptable purpose for use).
- P_{der}: authorised derivation purposes — set of pairs (data category, acceptable purpose for derivation).

Purposes are assumed to be taken from a fixed ontology, i.e. a centralised taxonomy.

Provenance Events (1)

- Usage policies defined above allow to defined requirements that must be fulfilled by components when processing data
- Compliance with policies can be evaluated using provenance data, but for this, provenance records must be represented in a unified and sufficiently expressive (depending on exact compliance requirements — *accountability by design*) way
- Instead of usual graph visualisation, we model provenance records as sequences of discrete events

Provenance Events (2)

$(\texttt{Acquire}, \Theta, C, \pi, P, t)$ A set of data with a corresponding set of data categories Θ was collected by a component C for a set of specific purposes P. The values of the data do not appear in the event, only its categories. In addition, this event contains the usage policy π set for the collected data and a timestamp t.

$(\texttt{Use}, \Theta, C, \rho, R, s, e)$ Use of a set of data categories Θ, e.g. {Postal address, Age group} by a component C.

$(\texttt{Export}, \Theta, C, C', \pi, P, t)$ A set of data, with a corresponding set of data categories Θ, was sent by component C to component C' at time t.

Provenance Events (3)

$(\texttt{Link}, \theta, \theta', \theta'', C, \pi, \rho, R, t)$ At time t component C linked two data elements with categories θ and θ'. The result is a single piece of data with category θ'', now available to component C.

$(\texttt{Derive}, \theta, \theta', C, \pi, \rho, R, t)$ At time t, component C derived data with category θ' from data with category θ.

$(\texttt{ReqRemove}, \Theta, t)$ At time t, total deletion of the set of data categories Θ was requested. Removal requests are assumed to be sent simultaneously to all components relevant to the set of data categories under consideration.

$(\texttt{Remove}, \Theta, C, t)$ The set of data categories Θ were deleted by component C at time t.

Provenance Event Example

$\lambda =$ (Link, Frequency, Treatment, Risk, Institute, π, Statistic, Correlation study, 2016-05-20T12:14)

Component Institute links two pieces of data with respective categories Frequency and Treatment. The result of this linking is a piece of data with category Risk.

The usage policy now associated with Risk for the component Institute is π, which restricts how the research institute can use the newly linked data set, but is not relevant for this event.

The provided reason for this linking operation is Correlation study, meant to justify the included purpose designation Statistic.

Goals of Correctness and Compliance Formalisation

Having defined both usage policies and provenance events enables two types of checks:

- Provenance *correctness*: a number of conditions must be fulfilled for provenance information to be coherent, independently of any usage policy. Sanity checks are possible and can be seen as a category of minimal guarantees.
- *Compliance* of provenance with usage policies: the compliance of recorded provenance can be analysed with regard to the predefined policies.

Formalism

- To model correctness and compliance, formalism needed to reason about provenance logs
- Defined a number of functions: event type, event time, active component, controller set, associated data categories, descended data categories, relative strength of usage policies, associated usage policy for a given component, data category and log
- Example: relative strength of usage policies: policy π' is *stronger than or equal to* policy π if all following conditions hold: (1) $\pi'.\mathsf{D_g} \leq \pi.\mathsf{D_g}$; (2) $\pi'.\mathsf{D_{rf}} \leq \pi.\mathsf{D_{rf}}$; (3) $\pi'.\mathsf{Fw} = (\top, \varnothing) \ \vee \ \pi'.\mathsf{Fw} = \pi.\mathsf{Fw} = (\bot, \varnothing) \ \vee \ \big(\pi'.\mathsf{Fw} = (\mathsf{bl}, \mathsf{List}') \ \wedge \ \pi.\mathsf{Fw} = (\mathsf{bl}, \mathsf{List}) \ \wedge \ \mathsf{List} \subseteq \mathsf{List}'\big) \ \vee \ \big(\pi'.\mathsf{Fw} = (\mathsf{wl}, \mathsf{List}') \ \wedge \ \pi.\mathsf{Fw} = (\mathsf{wl}, \mathsf{List}) \ \wedge \ \mathsf{List}' \subseteq \mathsf{List}\big)$; (4) $\pi.\mathsf{Li} \subseteq \pi'.\mathsf{Li}$; (5) $\pi.\mathsf{De} \subseteq \pi'.\mathsf{De}$; (6) $\pi'.\mathsf{P_{use}} \subseteq \pi.\mathsf{P_{use}}$; (7) $\pi'.\mathsf{P_{der}} \subseteq \pi.\mathsf{P_{der}}$

Rules for Internal Correctness of Logs

- Correctness rules ensure the *internal consistency* of event logs and are independent of the associated usage policy.
- 12 correctness rules Cor1–Cor12, detailed in paper. A log λ is said to be *correct* if all correctness rules hold for λ
- Example (Cor3): origin of data categories appearing in linking events:
  - $\lambda_i = (\texttt{Link}, \theta, \theta', \theta'', C, \pi, \rho, R, t) \Longrightarrow$ $\exists\, j, \Theta, C', \pi', \rho', P', t', R', \theta_1, \theta_2, \theta_3 \mid \big(\lambda_j = (\texttt{Acquire}, \Theta, C', \pi', P', t') \ \wedge\ \theta \in \Theta\big) \ \vee \lambda_j = (\texttt{Derive}, \theta_1, \theta, C', \pi', \rho', R', t') \ \vee\ \lambda_j = (\texttt{Link}, \theta_2, \theta_3, \theta, C', \pi', \rho', R, t') \ \wedge\ (t' < t)$
  - $\lambda_i = (\texttt{Link}, \theta, \theta', \theta'', C, \pi, \rho, R, t) \Longrightarrow$ $\exists\, j, \Theta, C', \pi', \rho', P', t', R', \theta_1, \theta_2, \theta_3 \mid \big(\lambda_j = (\texttt{Acquire}, \Theta, C', \pi', P', t') \ \wedge\ \theta' \in \Theta\big) \ \vee \lambda_j = (\texttt{Derive}, \theta_1, \theta', C', \pi', \rho', R', t') \ \vee\ \lambda_j = (\texttt{Link}, \theta_2, \theta_3, \theta', C', \pi', \rho, R', t') \ \wedge\ (t' < t)$

Rules for Compliance of Logs with Usage Policies

- Compliance rules depend on the values of associated usage policies
- 9 compliance rules Com1–Com9, detailed in paper. A log λ is said to be *compliant* if all compliance rules hold for λ
- Example (Com7): data categories in the associated derivation policy may neither be used to directly derive new data, nor indirectly:
  $\lambda_i = (\texttt{Derive}, \theta', \theta'', C, \pi, \rho, R, t) \ \wedge \ \theta' \in \mathrm{Dsc}(\lambda, \theta) \ \wedge$
  $\mathrm{EvTime}(\lambda_i) \geq \mathrm{EvTime}(\lambda_*(\lambda, \theta, C)) \Longrightarrow \theta \notin \pi_*(\lambda, \theta, C).\mathrm{De}$

Applicability to Privacy Accountability

- Privacy preferences are a matter of personal choice, hence one-size-fits-all approach impossible. Need to incorporate different trust models representing typical user perceptions
- Log itself may become privacy threat. Even metadata (data categories) can be sensitive. Secure log storage solutions must be used
- Limited guarantees about trustworthiness of provenance records when mapping to actual system events loose. External pressure from data processing authorities necessary
- No centralised purpose ontology exists, hence risk of abusive purpose designations by components if no oversight exists

Conclusion

- We introduced a formal correctness and compliance framework for provenance, based on a linear view of provenance events and the definition of sticky usage policies
- Presented formats for policies and rules for correctness and compliance are not meant to be exhaustive or applicable to all scenarios — this is just one instantiation
- Framework serves as a basis allowing to include other policy components and event types
- Open question: how can the global provenance record be aggregated securely from the different involved components when they are not all trusted?