Inference-Proof Monotonic Query Evaluation and View Generation Reconsidered

Joachim Biskup

▶ To cite this version:

Joachim Biskup. Inference-Proof Monotonic Query Evaluation and View Generation Reconsidered. 34th IFIP Annual Conference on Data and Applications Security and Privacy (DBSec), Jun 2020, Regensburg, Germany. pp.79-99, 10.1007/978-3-030-49669-2_5 . hal-03243627

HAL Id: hal-03243627
https://inria.hal.science/hal-03243627

Submitted on 31 May 2021

Inference-Proof Monotonic Query Evaluation and View Generation Reconsidered

Joachim Biskup

Fakultät für Informatik, Technische Universität Dortmund, Germany
joachim.biskup@cs.tu-dortmund.de

Abstract. The concept of inference-proofness has been introduced for capturing strong confidentiality requirements—including privacy concerns—of an information owner, communicating with a semi-honest partner by means of their message exchanging computing agents according to some agreed interaction protocols. Such protocols include closed-query evaluation and view generation by the information system agent under the control of the information owner, and the corresponding request preparation by the client agent. The information owner employs a security mechanism for controlled interactions, shielding the epistemic state of the information system agent and suitably altering messages sent to the client agent. The alterings provably guarantee that the partner cannot infer the validity of any piece of information that the information owner has declared as being prohibited. Based on selected previous work, we carefully describe and inspect the underlying function and attack scenario and summarize and analyze basic approaches for controlled interactions within an abstract framework for epistemic states.

Keywords: Abstract data source, A priori knowledge, Best current view, Closed-query evaluation, Confidentiality, Epistemic state, Inference-proofness, Interaction protocol, Lying, Prohibition, Security invariant, Simulated current view, View generation.

1 Introduction

High level security requirements like availability, integrity and confidentiality have been refined in sophisticated guidelines for constructing and evaluating secure computing systems of various kinds and, correspondingly, a rich variety of specific security mechanisms have been developed. Accordingly, for each concrete class of applications, in the spirit of computing engineering in general, a comprehensive range of considerations is due, from a precise specification of the wanted system functionality and the explicit description of the in most cases conflicting security interests of the expected user as well as of further "attackers" over mathematical models and their formal verification to final actual implementations and their ongoing multi-literal inspections.

This work is devoted to contribute to such a comprehensive view of security by reconsidering a specific kind of security mechanisms proposed to support the *confidentiality* interests as *exceptions* of the *availability* interests of an information owner while using an *information system* for *query evaluation* and *view generation* to communicate with some cooperation partner. Clearly, within this short article we again have to focus on aspects held to be particularly important, including the followings ones. What precisely is the object of protection? Who precisely is seen as an attacker and which precise means are exploited by him for violations? How to formally model the wanted kind of confidentiality? What kind of enforcing security mechanisms have been designed? How to mathematically verify their actual achievements? More concretely, we treat these concern by reconsidering a specific fraction of the in the meantime highly ramified line of research about confidentiality-preserving query–response interactions of a logic-oriented information system like a suitably restricted relational database system.

Even more specifically, our contributions can be summarized as follows, while the overall achievements and limitations are discussed in the conclusions:

- On the layer of social cooperation mediated by computing agents, in Section 2, we identify the "epistemic state of an information system agent" as the actual object in need of protection against a class of most powerful attackers.
- On the layer of computing agents, in Section 3, we further elaborate a formal model of abstract data sources, which captures the relevant features of monotonic and complete information systems.
- On the layer of security specification, in Section 4, we adapt inference-proofness as strong confidentiality to the model of abstract data sources.
- On the layer of security enforcement, in Section 5, we present unified expositions and verification of security mechanisms in terms of that model.

These contributions are—unifying and partly extending—extracted from the seminal work [13,7] and further specific refinements [2,4,3,5,6], which are part of larger efforts [1]. Moreover, we note that our treatment of confidentiality is in the spirit of various other work, e.g., already early ones on statistical database security [8] and about non-interference of general program execution [9], together with the rich elaborations of follow-up studies, which for example are concisely surveyed in [10]. In contrast to some other work, we do not aim at "total confidentiality" but see confidentiality as an exception from availability and, thus, allow specifically declared information flows like for declassification [11] and, additionally, we want to construct enforcing mechanism in the sense of [12].

2 Function and Attack Scenario

Since ever, among many other activities, and in a closely intertwined manner, people reason as individuals by acquiring, structuring, keeping and exploiting *information* to make up their respective minds and behave as social creatures by *communicating* with others. With the advent of computing technologies, both individually dealing with information and socially communicating have been partly delegated to *computing agents.* On the one hand, the delegation should facilitate routine task or even enhance human capabilities. On the other hand, depending on the context, as delegators, individuals at their discretion or groups of them according to some socially accepted norm aim to still control the computing agents executing protocols as their delegatees, or at least the human delegators should appropriately configure the computing delegatees.

Being aware of the resulting reduction, we can simplifying map concepts of human reasoning and communication to the inference protocols and interaction protocols of their computing agents and, correspondingly, actually performed human activities to protocol-complying computing process executions. Under such a reduction, and even more simplifying, a group of human individuals is modeled to be complemented by a *multi-agent* computing configuration. In this model, each *human individual* controls a dedicated computing agent that, at least partly and by means of *protocol executions*, both deals with the information owned by that individual, in particular by internally deriving an epistemic state from a chosen information representation, and mediates the communications of that individual, in particular by sending and receiving *messages* according to one or more agreed interaction protocols.

Though, in principle, each individual can act in diverse roles and, correspondingly, each controlled computing agent can execute diverse protocols, we further specialize the model sketched above in focusing on only two individuals with their computing agents. One individual is seen as an *information owner* controlling an *information system agent*, and the other individual is treated as a cooperating *communication partner* employing a *client agent*. Moreover, to enable *cooperation*, in principle the information owner is willing to *share information* with the communication partner. However, complying with *privacy issues* or pursuing other *confidentiality requirements*, as an exception from sharing, the information owner might want to hide some specific *pieces of information*.

Summarizing the simplified model, we assume an overall *framework* with the eight *features* outlined in the following and visualized in Figure 1.

1. [Epistemic state of information system agent as single object of protection.]
   The human information owner does not deal with information processing and reasoning by himself but only provides the inputs to the information system agent under his *control*. At each point in time, that agent is internally deriving a formally defined *epistemic state*.
2. [Mediation of human communications by interacting computing agents.]
   Once having agreed on cooperation, the human information owner and his human communication partner do not communicate directly with each other, but only mediated by the computing agents under their respective control.
3. [Dedicated access permissions for information sharing.]
   As a normally initial input to his information system agent, independently of the actual epistemic state, the information owner has granted dedicated *access permissions* to his communication partner. That permissions declare that over the time the client agent of the partner may *interact* with the information system agent of the owner following some explicitly chosen *interaction protocols* that exclusively refer to the internal epistemic state of the information system agent (but, e.g., not to the physical mind of the information owner or any "real world" besides the multi-agent model).

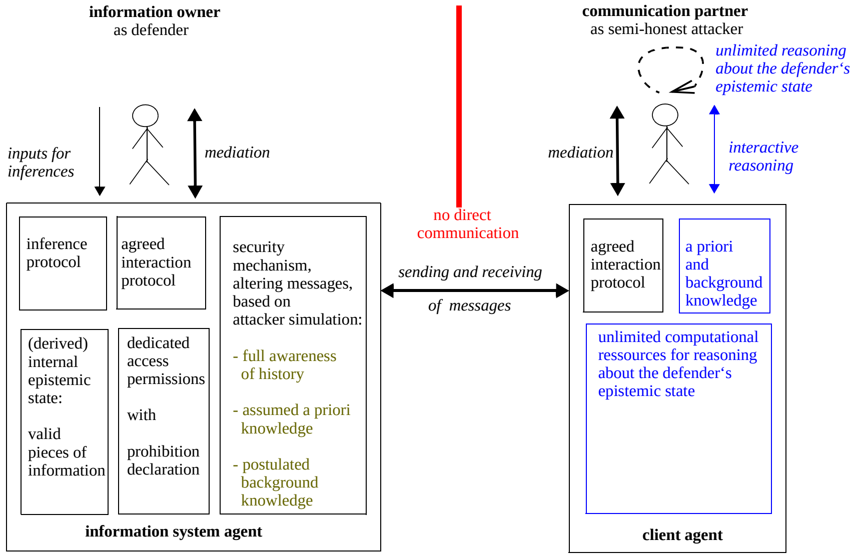
information owner
as defender
inputs for inferences
mediation
no direct communication
communication partner
as semi-honest attacker
unlimited reasoning about the defender‘s epistemic state
mediation
interactive reasoning
inference protocol
agreed interaction protocol
security mechanism, altering messages, based on attacker simulation:
- full awareness of history
- assumed a priori knowledge
- postulated background knowledge
(derived) internal epistemic state:
valid pieces of information
dedicated access permissions
with
prohibition declaration
information system agent
sending and receiving
of messages
agreed interaction protocol
a priori and background knowledge
unlimited computational ressources for reasoning about the defender‘s epistemic state
client agent

Fig. 1. The framework of a defending information owner with his information system agent and an attacking communication partner with his client agent

4. [Exceptions by explicit prohibitions designating pieces of information.]
   However, as a further normally initial input to his information system agent, also independently of the actual epistemic state, the information owner explicitly *declares* exceptions from the dedicated access permissions in the form of prohibitions. Each *prohibition* specifies a piece of information that the communication partner should not be able to learn. More precisely, each prohibition being expressed in terms of the information system agent and thus referring to possible epistemic states, the communication partner should never be able to become sure about the actual *validity* in the epistemic state of the information system agent. In other words, from the partner's point of view it should always appear to be *possible* that the prohibited piece of information is *not valid* in the epistemic state of the information system.
5. [Partner suspected to reason about validity of prohibitions.]
   Though the client agent is restricted to exactly follow the interaction protocols mentioned in the dedicated access permissions, the human communication partner can choose any sequence of permitted commands. Moreover, the communication partner is assumed to have unlimited computational resources when rationally reasoning about the validity or non-validity of a prohibited piece of information, whether employing the client agent under his control or any other means.
6. [Security mechanism implanted in owner's information system agent.]
   To enforce the confidentiality requirements of the information owner, the information system agent is enhanced by some *security mechanism* that should shield the underlying information processing from a direct contact with the

client agent. That security mechanism first inspects each message to be sent by the information system agent to the client agent according to the pertinent interaction protocol whether a *violation* of the information owner's confidentiality requirements would be enabled on the side of the communication partner. If this is the case, the security mechanism then *alters* the message such that the message is still as informative as possible on the one hand but all options for a violation are blocked on the other hand.

7. [Reasoning supported by a priori knowledge and background knowledge.]
   First of all, the communication partner's rational reasoning about the internal epistemic state of the information system agent is based on the messages exchanged by the respective computing agents and, thus, completely known to both agents. Additionally, the partner's rational reasoning is presumed to be supported by some *a priori knowledge* about the application dealt with in the cooperation between the two individuals involved and additional *background knowledge* comprising both a complete specification of the *interaction semantics* and the full *awareness of the security mechanism* (possibly even including the prohibition declaration) and, most notably, nothing else.
8. [Principle inaccessibility of the partner.]
   The internals of both the human communication partner and his client agent are considered to be principally inaccessible for the information owner and his system agent. This implies that the latter ones can only rely on *assumptions* about the details of the a priori knowledge and a *postulation* about the background knowledge available to the former ones.

We follow a somehow martial but common terminology of security in computing, which ignores that in many scenarios an individual involved as communication partner will primarily treated as cooperating in a friendly manner.

- Partially trusted for consciously sharing information in principle and correctly executing the agreed interaction protocols, the *communication partner* with the *client agent* is denoted as a *semi-honest attacker*, suspected to potentially aiming to maliciously infer the actual validity of pieces of information that the information owner has declared to be kept confidential.
- Accordingly, the *information owner*—together with the *information system agent* controlled by him—is denoted as the *defender*.

The security mechanism implanted in the defending information system agent has to invariantly enforce a suitable version of the following still informally expressed *security policy* of *inference-proofness*, which also specifies the *attacker model*: For each prohibited piece of information ψ, the information content of messages sent to the attacking client agent will never enable the attacking receiver to rationally infer that ψ is valid in the epistemic state, even when

- inspecting the complete *history* of preceding *interactions*,
- considering some *a priori knowledge* about the possible *epistemic states*,
- applying the *semantics* of the agreed *interaction protocols* and
- being *aware* of the *functionality* of the *security mechanism* (possibly even including the prohibition declaration).

The concept of *rationality* on the side of the attacker is then captured by the following rephrasement of the still to be suitably versioned security policy of *inference-proofness* in terms of *indistinguishability*:

> For each prohibited piece of information ψ, for each epistemic state d satisfying the a priori knowledge, for each sequence of messages exchanged during an interaction history and complying with the agreed interaction protocols but potentially altered by the security mechanism, there exists an "*alternative*" epistemic state d' such that (i) the same sequence of messages would be generated, in compliance with the agreed interaction protocols and subjected to the alterations by the security mechanism, but (ii) ψ is not valid in d'.

The epistemic state d is thought as actually be derived (or "stored") by the information system agent and might satisfy the prohibited piece of information ψ or not. The former case implies that the alternative state d' required to exist is different from d; in the latter case, the actually stored state d and the alternative state d' might be the same. Accordingly, declaring ψ as a prohibition does not intent to block any option to infer the non-validity of ψ.

Confidentiality as inference-proofness could be trivially achieved by granting no access permissions at all or altering the information content of all messages sent to the attacker to nothing, violating any conflicting *availability requirements* and, thus, making the whole thing useless. Accordingly, confidentiality requirements and availability requirements always have to be suitably balanced. We focus on the following three-level *conflict resolution* strategy, which leads to a combination of a *constraint solving problem* and an *optimization problem*:

1. As a general rule, for the sake of *availability*, some dedicated access permissions are granted, as far as not conflicting with level 2 of the strategy.
2. As exceptions, for the sake of *confidentiality* specific prohibitions are declared that have to be enforced by alterations made by the security mechanism, but to comply with level 3 of the strategy only as far as definitely necessary.
3. Again for the sake of *availability*, as limitation for the effect of exceptions, the alterations made have to be minimal.

3 Abstract Data Sources as Epistemic States

The notion of an *abstract information system* is intended to capture common important features of information contents like at least *semi-structured* and *logic-oriented knowledge bases*, including *relational databases* under suitable restrictions, and as far as they are *complete* regarding query answering under *monotonic reasoning*. A possibly occurring information content of such an abstract information system is seen as an *epistemic state*, though it is formally just an element of a pertinent finite or countably infinite set. We call such an element an "*abstract data source*", not assuming any internal properties.

However, we impose a set-algebraic structure with natural properties on the considered *universe* of all abstract data sources. These properties should reflect the *model-theoretic* approach of various monotonic logics to define *semantics* for the underlying syntax of a formal language by assigning *truth-values* to atomic sentences and then, by induction, to all sentences. In fact, if on the syntactic side the language provides means to express negation, conjunction and disjunction, then on the semantic side the corresponding *sets* of satisfying *truth-value assignments* (*models*, *interpretations*) are treated by (set) *complement*, (set) *intersection* and (set) *union*, respectively. So, using sentences to syntactically express *closed queries* whose semantics are the respective sets of satisfying truth-value assignments, we may identify a syntactic query expression with its semantic evaluation. Accordingly, totally refraining from any syntax for abstract information systems, we define queries as a particular sort of subsets of the universe.

For the case of an *infinite* universe with countably infinitely many queries, to deal with iteratively determined *fixpoints*, we even consider countably infinite intersections which, however, do not need to generate a query. Instead, we require *compactness* of the set of queries, which captures a straightforward corollary to the existence of a correct and complete proof system for classical first-order logic, roughly saying that any (possibly infinite) logical entailment implies a finite entailment (since formal proofs are finite by definition). Note that for the finite case this property trivially holds.

A (set) *inclusion* of the form $q_1 \subseteq q_2$ corresponds to a *logical entailment* in the logics that motivated our abstract settings. In fact, thinking of q_1 and q_2 as the satisfying sets of truth-value assignments for some sentences χ_1 and χ_2, respectively, then $q_1 \subseteq q_2$ says that each truth-value assignment that makes χ_1 true also does so for χ_2; this is just the classical definition of logical entailment.

Definition 1 (abstract data sources and closed queries).
A universe *of* data sources *is a (finite or infinite) set U.*
A query set *Q for a universe U satisfies the following properties:*

1. *$\{\emptyset, U\} \subseteq Q \subseteq \wp U$ with Q being finite or countably infinite;*
2. *Q is* closed *under* complement*,* finite *intersection and finite* union*;*
3. *Q is* compact*, i.e., for all $Q' \subseteq Q$, for all $q \in Q$,*
   if $\bigcap Q' \subseteq q$, then there exists a finite $Q'' \subseteq Q'$ such that $\bigcap Q'' \subseteq q$.

Moreover, $Seq(Q)$ is the set of all (possibly) infinite sequences *of queries and $Int(Q)$ is the set of all (possibly) infinite* intersections *of queries.*

Following the explanations and the common intuitive understanding, *query evaluation* could be defined for a query $q \in Q$ and a data source $d \in U$ by the expression IF $d \in q$ THEN *true* ELSE [IF $d \in U \backslash q$] *false*. However, for our formal investigations the following equivalent definition is more convenient.

Definition 2 (abstract (stepwise) query evaluation). *Let Q be a query set for an universe U. Then the* query evaluation *function is defined by*

$$quer : Q \times U \longrightarrow Q \quad \text{with} \quad quer(q, d) = \text{ IF } d \in q \text{ THEN } q \text{ ELSE } U \backslash q\,. \quad (1)$$

The stepwise extension $\quad quer : Seq(Q) \times U \longrightarrow Seq(Q) \quad$ *is defined by*

$$quer(\langle q_1, q_2, \ldots \rangle, d) = \langle quer(q_1, d), quer(q_2, d), \ldots \rangle. \quad (2)$$

Equation (1) immediately implies that for all data sources $d \in U$ and $d' \in U$, for all queries $q \in Q$ and $\tilde{q} \in Q$ the following assertions hold:

$$d \in quer(q, d)\,, \tag{3}$$

$$d' \in quer(q, d) \ \text{ iff } \ quer(q, d') = quer(q, d)\,, \tag{4}$$

$$\text{if } \ d \in q \ \text{ and } \ d' \in q \ \text{ then } \ q \subseteq quer(\tilde{q}, d) \ \text{ iff } \ q \subseteq quer(\tilde{q}, d')\,. \tag{5}$$

Besides being an element of the universe, the properties of a data source $d \in U$ are only determined by its query evaluations. In particular, two data sources are *indistinguishable* if they are contained in exactly the same queries. Hence, imagining an enumeration $q_1, q_2, \ldots,$ of all queries in Q, we can characterize a data source d as being in the intersection of its query evaluations. In this sense, the *best view* of d "from outside" is just this intersection, which always includes d itself but might also contain many further "indistinguishable" data sources. If the best view of d is a singleton, then it represents *complete* knowledge of d.

Definition 3 (abstract best view generation). *Let Q be a query set for an universe U. Then the* view generation *function is defined by*

$$view : U \longrightarrow Int(Q) \quad \textit{with} \quad view(d) = \bigcap_{q \in Q} quer(q, d)\,. \tag{6}$$

4 Inference-Proofness for Known Prohibition Declaration

As explained and motivated in Section 2 we imagine an owner of the abstract information system who implants a security mechanism into the defending information system agent under his control, aiming to enforce inference-proofness of interactions as a sophisticated kind of confidentiality regarding the message-based interactions with an attacking client agent operated by some only semi-honest communication partner. We study two interactions:

- *closed-query evaluation* with *response* preparation: we see the queries as request messages sent from the attacking client agent and the corresponding responses as reaction messages returned by the defending system agent;
- *view generation*: we image a (formally not represented) request message from the attacking client agent to obtain a best view and we see the generation result as the respond message returned by the defending system agent.

Besides the request messages and the respond messages, the formal notions of inference-proofness depend on two further parameters, to be declared by the information system owner specifically for the attacking client agent: (i) a set of *prohibitions*, i.e., pieces of information that the rationally reasoning attacker should not be able to learn; (ii) the *a priori knowledge* held by the attacker about the actually stored abstract data source, as assumed by the defender. The attacker is also implicitly postulated to be fully aware of the security mechanism employed by the defender and to even know the declared prohibitions.

The notion of the attacker's rationality is implicitly related to the semantics of query evaluation and view generation. Finally, our notions of inference-proofness include the following natural *security preconditions*:

- The stored data source complies with the (assumed) a priori knowledge.
- The (assumed) a priori knowledge does not violate the pertinent confidentiality requirement expressed by the prohibition declaration.

Definition 4 (abstract prohibitions and abstract a priori knowledge). *1. A* prohibition *is a query $p \in Q$, and a set of* enforceable[1] prohibitions *is denoted by $P \subseteq Q$. A* prohibition declaration *is a finite set $proh \subseteq P$.*
2. An assumed a priori knowledge *is a query $prior \in Q$, and a set of* tolerable[1] (pieces of) assumed a priori knowledge *is denoted by $A \subseteq Q$.*

The following four versions of inference-proofness formally reflect the intuitive assumption that the attacker *knows the prohibition declaration* by treating it being invariant under alternative data sources. Regarding controlled query evaluation, the wanted security mechanisms are naturally intended to operate *stepwise* and *history-dependent* but *without a look-ahead.*

Definition 5 (inference-proofness for closed-query evaluation).
1. $con\_quer : Seq(Q) \times U \times A \times \wp P \longrightarrow Seq(Q)$ is a stepwise controlled query evaluation *function iff for each (point in time) $t = 1, 2, \ldots$ the result value $con\_quer(\langle q_1, q_2, \ldots \rangle, d_{st}, prior, proh)_t$ depends in addition to d_{st}, prior and proh only on the finite prefix $q_1, q_2, \ldots, q_t$ of the argument sequence and, thus, implicitly, also on the finite prefix $r_1, r_2, \ldots, r_{t-1}$ of the result sequence with $r_i = con\_quer(\langle q_1, q_2, \ldots \rangle, d_{st}, prior, proh)_i$, such that we can define $con\_quer(\langle q_1, q_2, \ldots, q_t \rangle, d_{st}, prior, proh) = \langle r_1, r_2, \ldots, r_t \rangle$.*

2. The function is inference-proof *iff for each prohibition declaration $proh \in \wp P$, for each a priori knowledge $prior \in A$ such that $prior \nsubseteq p$ for all $p \in proh$, for each prohibition $p \in proh$, for each ("stored") data source $d_{st} \in prior$, for each sequence $\langle q_1, q_2, \ldots \rangle$ of closed queries $q_i \in Q$,*
there exists an ("alternative") data source $d^p_{al} \in prior$ such that

- indistinguishability *of d_{st} and d^p_{al} (w.r.t. the prohibition p): $con\_quer(\langle q_1, q_2, \ldots \rangle, d_{st}, prior, proh) = con\_quer(\langle q_1, q_2, \ldots \rangle, d^p_{al}, prior, proh)$;*
- harmlessness *of d_{al} (w.r.t. the prohibition p): $d^p_{al} \notin p$.*

3. The function is strongly inference-proof *iff it is inference-proof for proh substituted by $\{\bigcup proh\}$.*

Definition 6 (inference-proofness for view generation).
1. $con\_view : U \times A \times \wp P \longrightarrow Int(Q)$ is a controlled view generation *function.*

2. The function is inference-proof *or* strongly inference-proof*, respectively, iff the corresponding condition, but without the clause for the sequence of queries, of Definition 5, part 2. or part 3, respectively, holds.*

[1] For each practical framework, the notions of "enforceable" and "tolerable" have to be appropriately defined to capture application needs and complexity issues.

5 Controlled Interactions

We study three fundamental approaches to provably achieve inference-proofness for interaction sequences of unlimited length only consisting of closed-query evaluations together with the respective response preparations for the case of abstract data sources as epistemic states. The achievement of inference-proofness for sequences of stepwise *closed-query evaluations* will be based on enforcing a pertinent *security invariant* for all points in time t, starting with some pertinent *security precondition*. In the definitions of inference-proofness, the latter assertions are already formally stated and the former assertions are suggested by requiring the existence of a harmless data source. In fact, the attacker's *best current view bestcv* on the defender's actual data source d_{st} always consists of all those data sources d' that are *indistinguishable* from the actual one and, thus, constitutes the *least uncertainty* left to the attacker so far. Accordingly, the harmless data source d_{al} required to exist has to be an element of the attacker's best current view.

Conceptually, at each point in time t, the best current view is determined as a kind of an *inverse image* of the interaction history performed so far, i.e., of the submitted queries $q_1, q_2, \ldots, q_t$ under the a priori knowledge *prior* and the prohibition declaration *proh* together with the returned responses $con\_quer(\langle q_1, q_2, \ldots, q_t \rangle, d_{st}, prior, proh) = \langle r_1, r_2, \ldots, r_t \rangle$. More formally, for the *best current view* we have the following:

$$\begin{aligned} bestcv_t = & \{\, d \mid d \in U \cap prior, \text{ and } con\_quer(\langle q_1, q_2, \ldots, q_t \rangle, d_{st}, prior, proh) \\ & \qquad\qquad\qquad = con\_quer(\langle q_1, q_2, \ldots, q_t \rangle, d, prior, proh) \,\} \\ = & con\_quer^{-1}[con\_quer(\langle q_1, q_2, \ldots, q_t \rangle, d_{st}, prior, proh)] \cap prior\,; \end{aligned}$$

$$\begin{aligned} bestcv_{\infty} = & \{\, d \mid d \in U \cap prior, \text{ and } con\_quer(\langle q_1, q_2, \ldots \rangle, d_{st}, prior, proh) \\ & \qquad\qquad\qquad = con\_quer(\langle q_1, q_2, \ldots \rangle, d, prior, proh) \,\} \\ = & con\_quer^{-1}[con\_quer(\langle q_1, q_2, \ldots \rangle, d_{st}, prior, proh)] \cap prior\,. \end{aligned}$$

As a technical means, however, a *security mechanism* might only maintain a *simulated current view simcv* still invariantly containing a harmless data source, which is employed for checking tentative updates of the attacker's uncertainty for violations of the security invariant. For studying abstract information systems refraining from representing syntax at all, we will use such a simulated current view directly as a kind of *log file* to keep the *essence* of the interaction history.

Though we are literally speaking about technical means having machine-executable programs in mind, we deal with abstract information systems as purely mathematical objects and, accordingly, do not actually care about computability. Nevertheless, by abuse of language, we will denote purely mathematical methods for controlled interactions as algorithms, since we have come up with even efficiently computable procedures for suitable refinements based on appropriate syntactic representations of the mathematical items.

5.1 Controlled Query Evaluation by Refusing

For the approach to alterations of a harmful query evaluation by *refusing*, the existence of an *"alternative" harmless data source* will explicitly be monitored by inspecting the assertion "for all $p \in proh$: $simcv \nsubseteq p$" as part of the security invariant enforced for each response to a submitted query. In fact, if a (previously unknown) *correct response* is returned to the attacker, then the invariance of the assertion after updating *simcv* accordingly has been confirmed explicitly by a tentative update before.

However, to additionally enforce the *indistinguishability* property to avoid meta-inferences from the fact of observing a refusal, we have to strengthen the invariant such that it becomes *independent* of the actual results of the query evaluations. In fact, if a (previously unknown) *correct response* is returned and the simulated current view *simcv* has actually been updated accordingly, then not only the tentative update with that response but also with its complement has been inspected for harmlessness explicitly before. Consequently, if at least one alternative has been found to be harmful, the resulting refusal might be caused by the correct response or its complement, such that the attacker cannot find out which alternative has actually occurred. For convenience, here *refusing* is signified be returning the universe U, which provides no new information, and, accordingly, no update of the simulated current view *simcv* is necessary.

As a special case, the correct response might *already be known* from the a priori knowledge together with the responses to previously inspected queries, as summarized in the value of the simulated current view *simcv*. To avoid an unnecessary refusal, this case is dealt with separately, by just confirming the correct query evaluation and, consequently, leaving *simcv* unchanged.

Theorem 1 (inference-proofness by refusing). *The stepwise controlled query evaluation function with alterations by refusing for a known prohibition declaration, as computed by Algorithm 1, is* inference-proof *(and* strongly inference-proof *under the substitution of proh by* $\{\bigcup proh\}$*).*

Proof. A full proof is given in the appendix. Here we only sketch the overall structure of the proof. An execution of Algorithm 1 determines a sequence of values for the simulated current view *simcv* with a fixpoint, such that

$$prior = simcv_0 \supseteq simcv_1 \supseteq simcv_2 \supseteq \ldots \text{ with } simcv_\infty = \bigcap_{t=0,1,2,\ldots} simcv_t \,.$$

By the construction and by assertion (3), $d_{st} \in simcv_\infty$. By an inductive argument based on the compactness, and because of the explicit check of the security invariant in step 10, we have $simcv_\infty \nsubseteq p$ for all $p \in proh$. Thus, for each $p \in proh$ there exists a data source $d^p_{al} \in simcv_\infty \backslash p$, which satisfies the precondition and is harmless by the construction.

Moreover, d^p_{al} is also *indistinguishable* (of the "stored" data source d_{st}), as is even any data source $\tilde{d} \in simcv_\infty$. Basically, this claim follows from the inductive procedure to decide whether the value of *simcv* should be changed, based on the *instance independent* security invariant enforced by step 10. □

```
   Input:  $\langle q_1, q_2, \ldots \rangle$   queue of closed queries, submitted by attacker
           $d_{st}$                         stored abstract data source
           $prior$                          a priori knowledge as query
           $proh$                           prohibition declaration as finite set of queries
   Output: $\langle r_1, r_2, \ldots \rangle$ list of (possibly) altered responses, returned to attacker

 1 $time \leftarrow 0$                      //initialize counter for discrete points in time;
 2 $simcv \leftarrow prior$                 //initialize simulated current view;
 3 repeat
 4     $time \leftarrow time + 1$;
 5     $query \leftarrow$ receive next query $q_{time}$ from input queue;
 6     $correct \leftarrow quer(query, d_{st})$          //determine correct query evaluation;
 7     if $simcv \subseteq correct$ then
 8         return $correct$ to output list
             //confirm correct response; leave $simcv$ unchanged
 9     else
10         if for all $p \in proh$: $simcv \cap query \nsubseteq p$  and  $simcv \cap (U \backslash query) \nsubseteq p$
             then
11             return $correct$ to output list      //respond correctly;
12             $simcv \leftarrow simcv \cap correct$
                 //update simulated current view accordingly
13         else
14             return $U$ to output list
                 //signify refusing; leave $simcv$ unchanged
15         end
16     end
17 until input queue has externally been closed, if ever;
```

Algorithm 1: Stepwise controlled query evaluation with alterations by refusing for a known prohibition declaration

Theorem 2 (refusing provides best current view directly). *Algorithm 1 executed for inputs $\langle q_1, q_2, \ldots \rangle$, d_{st}, prior and proh satisfying the preconditions $d_{st} \in prior$ and $prior \nsubseteq p$ for all $p \in proh$ for inference-proofness provides the* best current view *$bestcv_\infty$ by the fixpoint $simcv_\infty$ of the* simulated current view *simcv, i.e., we have $bestcv_\infty = simcv_\infty$.*

Proof. In the proof of Theorem 1 we show that $bestcv_\infty \supseteq simcv_\infty$. Conversely, assume $\tilde{d} \in prior$ but $\tilde{d} \notin simcv_\infty$. Then executing the Algorithm 1 for d_{st} and $\tilde{d}$, respectively, yields the same value for *simcv* at time 0 according to step 2 but different values for some later point in time. Consider the point in time $min > 0$ such that for the first time the executions differ for the value *simcv*. Accordingly, at time *min* for at least one of the data sources there was no refusing and, by the independence of the guarding expression in step 10, for both of them there was no refusing. Moreover, by the minimality of *min*, the query evaluations have been different, i.e., $quer(query_{min}, d_{st}) \neq quer(query_{min}, \tilde{d})$, such that the executions can be distinguished. Hence $\tilde{d} \notin bestcv_\infty$. □

5.2 Controlled Query Evaluation by Lying

For alterations by *lying*, the existence of an *"alternative" harmless data source* has to be ensured regarding a strengthen version of harmlessness that requires non-elementship in the *union over the prohibition declaration*. This version avoids the *hopeless situation* arising from lies on both that union and all its contributing prohibitions. The existence of such a data source will only partly explicitly be monitored, aiming to make the assertion "$simcv \not\subseteq \bigcup proh$" part of the security invariant. In fact, if a *correct response* is returned to the attacker, then the invariance of the assertion after updating *simcv* accordingly has been checked explicitly by a tentative update before. Otherwise, if a *lied response* is returned to the attacker, then no explicit additional inspection is necessary. Moreover, the *indistinguishability* property is also already implicitly be enforced, since a data source that satisfies each of the responses generated for the actual data source—whether correct or lied— turns out to generate the same reactions.

Theorem 3 (strong inference-proofness by lying). *The stepwise controlled query evaluation function with alterations by lying, as computed by Algorithm 2, is* strongly inference-proof.

Proof. Structurally as for refusing, by an inductive argument that the correct response and the lied response are not both harmful for a single prohibition. □

```
Input:  $\langle q_1, q_2, \dots \rangle$   queue of queries, submitted by attacker
        $d_{st}$                               stored abstract data source
        $prior$                                a priori knowledge as query
        $proh$                                 prohibition declaration as finite set of queries
Output: $\langle r_1, r_2, \dots \rangle$ list of (possibly) altered responses, returned to attacker

$time \leftarrow 0$                 //initialize counter for discrete points in time;
$simcv \leftarrow prior$            //initialize simulated current view;
repeat
    $time \leftarrow time + 1$;
    $query \leftarrow$ receive next query $q_{time}$ from input queue;
    $correct \leftarrow quer(query, d_{st})$        //determine correct query evaluation;
    $lied \leftarrow U \backslash correct$          //prepare the lie;
    if $simcv \cap correct \not\subseteq \bigcup proh$ then
        return $correct$ to output list     //respond correctly;
        $simcv \leftarrow simcv \cap correct$
          //update simulated current view accordingly
    else
        return $lied$ to output list        //respond by the lie;
        $simcv \leftarrow simcv \cap lied$
          //update simulated current view accordingly
    end
until $input\ queue\ has\ externally\ been\ closed,\ if\ ever$;
```

Algorithm 2: Stepwise controlled query evaluation with alterations by lying for a known prohibition declaration

5.3 Controlled Query Evaluation by Combination

For alterations by a *combination of refusing and lying*, the existence of an *"alternative" harmless data source* will explicitly be monitored by inspecting the assertion "for all $p \in proh$: $simcv \not\subseteq p$" as the security invariant. In fact, first the *correct response* is explicitly inspected for harmlessness by a tentative update of *simcv*, and only in case of a failure, subsequently the *lied response* is also explicitly inspected for harmlessness. If both inspections fails, i.e., both the correct response and the lied response are harmful, then refusing is due. No further means are necessary to achieve the *indistinguishability* property as well.

Theorem 4 (inference-proofness by combination). *The stepwise controlled query evaluation function with alterations by a combination of refusing and lying, as computed by Algorithm 3, is* inference-proof *(and* strongly inference-proof *under the substitution of proh by* $\{\bigcup proh\}$*).*

Proof. Similar as for the proof of Theorem 1, following its overall structure. □

```
Input:  $\langle q_1, q_2, \ldots \rangle$   queue of queries, submitted by attacker
        $d_{st}$                 stored abstract data source
        $prior$                  a priori knowledge as query
        $proh$                   prohibition declaration as finite set of queries
Output: $\langle r_1, r_2, \ldots \rangle$ list of (possibly) altered responses, returned to attacker

$time \leftarrow 1$                //initialize counter for discrete points in time;
$simcv \leftarrow prior$           //initialize simulated current view;
repeat
    $time \leftarrow time + 1$;
    $query \leftarrow$ receive next query $q_{time}$ from input queue;
    $correct \leftarrow quer(query, d_{st})$        //determine correct query evaluation;
    $lied \leftarrow U \backslash correct$              //prepare the lie;
    if for all $p \in proh$: $simcv \cap correct \not\subseteq p$ then
        return $correct$ to output list      //respond correctly;
        $simcv \leftarrow simcv \cap correct$
          //update simulated current view accordingly
    else
        if for all $p \in proh$: $simcv \cap lied \not\subseteq p$ then
            return $lied$ to output list      //respond by the lie;
            $simcv \leftarrow simcv \cap lied$
              //update simulated current view accordingly
        else
            return $U$ to output list
              //signify refusing and leave $simcv$ unchanged
        end
    end
until input queue has externally been closed, if ever;
```

Algorithm 3: Stepwise controlled query evaluation with alterations by a combination of refusing and lying for a known prohibition declaration

5.4 Controlled View Generation

So far, we have studied stepwise controlled query evaluation functions for abstract data sources as epistemic states, employing refusing or lying or the combination of refusing and lying, respectively, as alterations of a harmful query evaluation. These functions are proven to be inference-proof for any sequence of closed-query evaluation with response preparation. Each proof has been based on investigating the properties of the sequence of the *simulated current views* maintained by the pertinent algorithm to keep track of the *interaction history* and to enforce a suitable *security invariant*, together with the fictitious fixpoint of that sequence. Essentially, this fixpoint is the intersection of the (possibly) altered query responses.

We further exploit the three fundamental approaches for such an algorithm to deal with the interaction of *view generation*. A *best view* is abstractly defined as the intersection of the query evaluations of *all* queries in the considered query set. Then, roughly outlined, we can form a queue of *all* such queries, or a suitably *exhaustive* part of it, submit it to the pertinent algorithm, and will (at least conceptually) obtain the fixpoint as a (possibly) altered inference-proof view. In an interaction of *controlled view generation*, that fixpoint can be returned to the communication partner suspected to be only semi-honest and attacking the dedicated prohibition declaration.

Theorem 5 (inference-proofness by refusing, lying and the combination). *The controlled view generation functions with alterations by refusing or lying or the combination of refusing and lying, respectively, for a known prohibition declaration, as computed by Algorithm 4, are* weakly or strongly or weakly inference-proof, *respectively.*

Proof. The claim straightforwardly follows from the inference-proofness of the imported algorithms. □

```
Input:  $d_{st}$      stored abstract data source
        $prior$      a priori knowledge as query
        $proh$       prohibition declaration as finite set of queries
Output: $view$       returned to attacker as controlled view

Import: Algorithm $i$ for
  either $i = 1$: refusing or $i = 2$: lying or $i = 3$: combination

form exhaustive queue $\langle q_1, q_2, \dots \rangle$ of closed queries;
apply Algorithm $i$ to $\langle q_1, q_2, \dots \rangle$ and the inputs, using local variable $simcv$;
on exit from the repeat-loop (actually or fictitiously) do
$view \leftarrow \bigcap simcv$;
return $view$ as output
```

Algorithm 4: Controlled view generation with alterations by refusing, lying or the combination of refusing and lying based on Algorithm 1, Algorithm 2 or Algorithm 3, respectively, for a known prohibition declaration

5.5 Some Comparisons

For refusing, we have $d_{st} \in simcv$ and $simcv = bestcv$. Basically, this intuitively means that the literal claims of returned controlled responses are a correct and complete disjunctive weakening of the best view. In contrast, for lying and the combination, whenever a lied response has actually occurred, we have $d_{st} \notin simcv$ and, consequently, $simcv \neq bestcv$, in particular saying that literal claims do not directly reflect the actual situation. The following theorem shows that this difference disappears under the substitution of *proh* by $\{\bigcup proh\}$, since then the occurrence of a refusal corresponds to a potential lie.

Theorem 6 (best current views for aggregated policy declaration). *Under the substitution of proh by* $\{\bigcup proh\}$*, the inverse functions of the controlled view generation functions with alterations by refusing or lying or the combination of refusing and lying, respectively, as computed by Algorithm 4, yield the same best current views.*

Proof. The full proof, given in the appendix, shows that for the single aggregated prohibition in $\{\bigcup proh\}$, the effect of the instance-independent check for harmfulness by refusing corresponds to the effect of instance-independently always returning a harmless response by lying. □

6 Conclusions

Enforcing inference-proofness as a sophisticated version of confidentiality relies on crucial assumptions about the a priori knowledge of the specific attacker and further postulations about the overall attack scenario. Furthermore, the notion of a defender or an attacker, respectively, refers to both human individuals and the computing agents under their control. Accordingly, for coming up with formally provable assertions about confidentiality the precise specification of the object to be protected by a security mechanism on the defender side as well as a precise specification of the capabilities on the attacker side are mandatory.

Our main contributions are complying with these requirements. On the defender side the epistemic state of the information system agent is identified as the basic protection object, independently of the actual syntactic representation and of any additional knowledge held by the human information owner. On the attacker side, our characterization of the attacker as a rational reasoner about message observations, a priori knowledge, the semantics of the agreed interactions and the functionality of the security mechanism refers to both the client agent and the human communication partner. Accordingly, the defender side is restricted by the possibilities of efficient algorithms, whereas the attacker side might employ unlimited resources. However, as far as the attacker relies on the computing resources of the client agent, refusing and lying essentially differ in determining the best current view: while for refusing the best current view is directly delivered by the returned accumulated information represented by the simulated current view, for lying the best current view has to be generated by a sophisticated function inversion procedure.

We have focused on conceptual and computational foundations rather than on specific applications. Regarding usability, our foundational results suggest that in each concrete practical situation we might be forced to admit relaxations and approximations. Regarding computational complexity, view generation as an off-line procedure might be preferred to closed-query evaluations as a dynamic and often time-constrained protocol. Moreover, in practice we are faced with structured epistemic states which allow more sophisticated interactions, e.g., open (SQL-like) queries and update transactions for relational databases. Interactions might also refer to non-monotonic operations regarding a structured epistemic state seen as "belief", e.g., a revision under suitable postulates. So far, these and further issues have already been preliminarily studied for specific frameworks, as discussed in [1]. It would be worthwhile to unify and further elaborate all these studies as an enhancement and extension of the present work.

References

1. Biskup, J.: Selected results and related issues of confidentiality-preserving controlled interaction execution. In: Gyssens, M., Simari, G.R. (eds.) 9th International Symposium on Foundations of Information and Knowledge Systems, FoIKS 2016. Lecture Notes in Computer Science, vol. 9616, pp. 211–234. Springer (2016)
2. Biskup, J., Bonatti, P.A.: Lying versus refusal for known potential secrets. Data Knowl. Eng. 38(2), 199–222 (2001)
3. Biskup, J., Bonatti, P.A.: Controlled query evaluation for enforcing confidentiality in complete information systems. Int. J. Inf. Sec. 3(1), 14–27 (2004)
4. Biskup, J., Bonatti, P.A.: Controlled query evaluation for known policies by combining lying and refusal. Ann. Math. Artif. Intell. 40(1-2), 37–62 (2004)
5. Biskup, J., Bonatti, P.A., Galdi, C., Sauro, L.: Optimality and complexity of inference-proof data filtering and CQE. In: Kutylowski, M., Vaidya, J. (eds.) 19th European Symposium on Research in Computer Security, ESORICS 2014, Part II. Lecture Notes in Computer Science, vol. 8713, pp. 165–181. Springer (2014)
6. Biskup, J., Bonatti, P.A., Galdi, C., Sauro, L.: Inference-proof data filtering for a probabilistic setting. In: Brewster, C., Cheatham, M., d'Aquin, M., Decker, S., Kirrane, S. (eds.) 5th Workshop on Society, Privacy and the Semantic Web – Policy and Technology, PrivOn2017. CEUR Workshop Proceedings, vol. 1951. CEUR-WS.org (2017), `http://ceur-ws.org/Vol-1951/PrivOn2017\_paper\_2.pdf`
7. Bonatti, P.A., Kraus, S., Subrahmanian, V.S.: Foundations of secure deductive databases. IEEE Trans. Knowl. Data Eng. 7(3), 406–422 (1995)
8. Denning, D.E.: Cryptography and Data Security. Addison-Wesley, Reading, MA (1982)
9. Goguen, J.A., Meseguer, J.: Unwinding and inference control. In: IEEE Symposium on Security and Privacy. pp. 75–87 (1984)
10. Halpern, J.Y., O'Neill, K.R.: Secrecy in multiagent systems. ACM Trans. Inf. Syst. Secur. 12(1), 5.1–5.47 (2008)
11. Sabelfeld, A., Sands, D.: Declassification: Dimensions and principles. Journal of Computer Security 17(5), 517–548 (2009)
12. Schneider, F.B.: Enforceable security policies. ACM Trans. Inf. Syst. Secur. 3(1), 30–50 (2000)
13. Sicherman, G.L., de Jonge, W., van de Riet, R.P.: Answering queries without revealing secrets. ACM Trans. Database Syst. 8(1), 41–59 (1983)

Acknowledgements: I would like to sincerely thank Piero Bonatti very much for fruitful cooperation on the works underlying this article.

Appendix 1: Proof of Theorem 1

Consider the execution of Algorithm 1 for some inputs $\langle q_1, q_2, \ldots \rangle$, d_{st}, *prior* and *proh* satisfying the preconditions $d_{st} \in \mathit{prior}$ and $\mathit{prior} \nsubseteq p$ for all $p \in \mathit{proh}$. Let $\langle \mathit{simcv}_0, \mathit{simcv}_1, \mathit{simcv}_2, \ldots \rangle$ be the sequence of values obtained by the simulated current view *simcv*, with $\mathit{simcv}_0 = \mathit{prior}$ according to step 2, and with $\mathit{simcv}_{\mathit{time}}$ being the updated value at the end of the *time*-th iteration of the repeat-loop for $\mathit{time} > 0$, according to either step 12, 14 or 8, respectively. Then we have

$$\mathit{simcv}_0 \supseteq \mathit{simcv}_1 \supseteq \mathit{simcv}_2 \supseteq \ldots . \qquad (7)$$

Define the fixpoint of this chain as

$$\mathit{simcv}_\infty = \bigcap_{\mathit{time}=0,1,2,\ldots} \mathit{simcv}_{\mathit{time}}. \qquad (8)$$

This fixpoint has the following properties:

1. $\mathit{simcv}_\infty \in \mathit{Int}(Q)$, according to Definition 1.
2. $d_{st} \in \mathit{simcv}_\infty$, by the *construction* during the execution of Algorithm 1, since for each $\mathit{time} = 0, 1, 2, \ldots$ one of the following alternatives apply: $d_{st} \in \mathit{prior}$ in step 2; $d_{st} \in \mathit{correct}$ in step 8 or 12 by assertion (3); $d_{st} \in U$ in step 14.
3. $\mathit{simcv}_\infty \nsubseteq p$ for all $p \in \mathit{proh}$, based on the *compactness* of the query set Q according to Definition 1, as verified below.

Let $p \in \mathit{proh}$. Assume indirectly that $\mathit{simcv}_\infty \subseteq p$. Then, by the compactness of the query set Q, there would exist a finite set F of values in the sequence (7) having a minimal element simcv_F (with maximum index of time) such that $\mathit{simcv}_F = \bigcap F \subseteq p$. Let then *min* be the first time such that $\mathit{simcv}_{\mathit{min}} \subseteq p$. By the precondition, $\min > 0$. Then, depending on the evaluation of the guarding expressions in step 7 and step 10, respectively, either $\mathit{simcv}_{\mathit{min}} = \mathit{simcv}_{\mathit{min}-1} \cap \mathit{correct}_{\mathit{min}}$ according to step 12 in the inner if-branch or $\mathit{simcv}_{\min} = \mathit{simcv}_{\min-1}$ according to step 14 in the inner else-branch or $\mathit{simcv}_{\min} = \mathit{simcv}_{\min-1}$ according to step 8 in the outer if-branch. However, the first case contradicts the value of the inner guarding expression and the second case and third case contradict the definition of *min*.

So, by $\mathit{simcv}_\infty \nsubseteq p$ there exists a data source $d^p_{al} \in \mathit{simcv}_\infty \backslash p$. We claim that d^p_{al} is the "alternative" data source required to exist:

4. d^p_{al} *satisfies the a priori knowledge*, since $d^p_{al} \in \mathit{simcv}_\infty \backslash p \subseteq \mathit{simcv}_\infty \subseteq \mathit{simcv}_0 = \mathit{prior}$.
5. d^p_{al} *is harmless* (w.r.t. the prohibition p), i.e., $d^p_{al} \notin p$, by the construction.
6. d^p_{al} *is indistinguishable* (of the "stored" data source d_{st}), since below we can show by induction that for each $\mathit{time} = 1, 2, \ldots$ the repeat-loop of Algorithm 1 takes the same actions, in fact not only for d^p_{al} but even for all data sources $\tilde{d} \in \mathit{simcv}_\infty$.

So, consider any $\tilde{d} \in \mathit{simcv}_\infty$ and suppose inductively that the value $\mathit{simcv}_{\mathit{time}-1}$ of the simulated current view *simcv* is the same for the stored data source d_{st} and the considered data source $\tilde{d}$.
Case 1: $\mathit{simcv}_{\mathit{time}-1} \subseteq \mathit{quer}(\mathit{query}_{\mathit{time}}, d_{st})$.
Then, the outer guarding expression at step 7 is true for d_{st} and, thus, the response $\mathit{quer}(\mathit{query}_{\mathit{time}}, d_{st})$ is returned in step 8. Then we have

$$\tilde{d} \in \mathit{simcv}_\infty \subseteq \mathit{simcv}_{\mathit{time}-1} \subseteq \mathit{quer}(q_{\mathit{time}}, d_{st})$$

and thus, by assertion (4), $\mathit{quer}(q_{\mathit{time}}, \tilde{d}) = \mathit{quer}(q_{\mathit{time}}, d_{st})$. This equality implies that also $\mathit{simcv}_{\mathit{time}-1} \subseteq \mathit{quer}(\mathit{query}_{\mathit{time}}, \tilde{d})$ and, accordingly, that the outer guarding expression at step 7 is also true for $\tilde{d}$ such that $\mathit{quer}(\mathit{query}_{\mathit{time}}, \tilde{d})$ is returned in step 8 as the same response.
Case 2: $\mathit{simcv}_{\mathit{time}-1} \not\subseteq \mathit{quer}(\mathit{query}_{\mathit{time}}, d_{st})$.
Case 2.1: For all $p' \in \mathit{proh}$: $\mathit{simcv}_{\mathit{time}-1} \cap \mathit{query}_{\mathit{time}} \not\subseteq p'$ and $\mathit{simcv}_{\mathit{time}-1} \cap (U \backslash \mathit{query}_{\mathit{time}}) \not\subseteq p'$.
Then $\mathit{quer}(\mathit{query}_{\mathit{time}}, d_{st})$ is returned in step 11 and *simcv* is updated accordingly in step 12, and we have

$$\tilde{d} \in \mathit{simcv}_\infty \subseteq \mathit{simcv}_{\mathit{time}} = \mathit{simcv}_{\mathit{time}-1} \cap \mathit{quer}(q_{\mathit{time}}, d_{st}) \subseteq \mathit{quer}(q_{\mathit{time}}, d_{st})$$

and thus, by assertion (4), $\mathit{quer}(q_{\mathit{time}}, \tilde{d}) = \mathit{quer}(q_{\mathit{time}}, d_{st})$. This equality implies that also $\mathit{simcv}_{\mathit{time}-1} \not\subseteq \mathit{quer}(\mathit{query}_{\mathit{time}}, \tilde{d})$ and, accordingly, that the outer guarding expression at step 7 is also false for $\tilde{d}$ and the inner guarding expression is checked in line 10. Being independent of the query evaluation, this expression is also true for $\tilde{d}$ by the assumption of Case 2.1, such that $\mathit{quer}(\mathit{query}_{\mathit{time}}, \tilde{d})$ is returned in step 11 as the same response and the same update of *simcv* occurs in step 12.
Case 2.2: For some $p' \in \mathit{proh}$: $\mathit{simcv}_{\mathit{time}-1} \cap \mathit{query}_{\mathit{time}} \subseteq p'$ or $\mathit{simcv}_{\mathit{time}-1} \cap (U \backslash \mathit{query}_{\mathit{time}}) \subseteq p'$.
Then the universe U is returned in step 14, signifying a refusal for d_{st}. Regarding $\tilde{d}$, since both $d_{st} \in \mathit{simcv}_{\mathit{time}-1}$ and $\tilde{d} \in \mathit{simcv}_{\mathit{time}-1}$, we have $\mathit{simcv}_{\mathit{time}-1} \not\subseteq \mathit{quer}(\mathit{query}_{\mathit{time}}, \tilde{d})$ by the assumption of Case 2 and assertion (5). So, the outer guarding expression in step 7 is also false for $\tilde{d}$ and the inner guarding expression is checked in line 10. Being independent of the query evaluation, this expression is also false for $\tilde{d}$ by the assumption of Case 2.2, such that the universe U is returned in step 14, signifying a refusal as the same response. □

Appendix 2: Proof of Theorem 6

By Theorem 2, we already know that for refusing with suitable inputs the simulated current view simcv_∞ equals the best current view bestcv_∞. Thus it suffices to show the following claim by induction:

> For a single aggregated prohibition $\bigcup \mathit{proh}$, the simulated current view $\mathit{simcv}_{\mathit{time}}$ of refusing equals the best current views $\mathit{bestcv}_{\mathit{time}}$ of lying and the combination, respectively.

At $time = 0$, for all of the three approaches we have $simcv_0 = bestcv_0$.

At $time > 0$, assume inductively that $simcv_{time-1}$ for refusing equals $bestcv_{time-1}$ for lying and the combination, respectively.

Case 1: Refusing returns the correct response $quer(query_{time}, d_{st})$.

Case 1.1: $simcv_{time-1} \subseteq quer(query_{time}, d_{st})$, i.e., refusing confirms the correct response. This response is harmless, for otherwise $simcv_{time-1}$ would already be harmful, contradicting the security invariant. Then, for refusing,

$$simcv_{time} = simcv_{time-1} = simcv_{time-1} \cap quer(query_{time}, d_{st})\,. \tag{9}$$

Case 1.2: Otherwise, we have for all $p \in proh$: $simcv \cap query \nsubseteq p$ and $simcv \cap (U \backslash query) \nsubseteq p$ and, again, for refusing,

$$simcv_{time} = simcv_{time-1} \cap quer(query_{time}, d_{st})\,. \tag{10}$$

In both subcases, regarding lying, the correct response $quer(query_{time}, d_{st})$ is returned for d_{st}, as exactly for all $d' \in quer(query_{time}, d_{st})$, for each of which we $quer(query_{time}, d') = quer(query_{time}, d_{st})$ by (4). The same reasoning applies for the combination. Accordingly, we have for both lying and the combination

$$bestcv_{time} = bestcv_{time-1} \cap quer(query_{time}, d_{st})\,, \tag{11}$$

together with (9), (10) and the induction assumption implying the claim.

Case 2: Refusing returns U to signify a refusal and, thus, for refusing,

$$simcv_{time} = simcv_{time-1} \cap U = simcv_{time-1}\,. \tag{12}$$

Then, according to the instance-independent guarding expression for refusing, there exists $p' \in \{\bigcup proh\}$ such that $simcv_{time-1} \cap query_{time} \subseteq p'$ or $simcv_{time-1} \cap (U \backslash query_{time}) \subseteq p'$. This implies that we have

either $simcv_{time-1} \cap query_{time} \subseteq \bigcup proh$
or $simcv_{time-1} \cap (U \backslash query_{time}) \subseteq \bigcup proh$
but not both.

For assume otherwise that both inclusions hold, then

$simcv_{time-1}$
$= simcv_{time-1} \cap U$
$= simcv_{time-1} \cap (query_{time} \cup (U \backslash query_{time}))$
$= (simcv_{time-1} \cap query_{time}) \cup (simcv_{time-1} \cap (U \backslash query_{time}))$
$\subseteq \bigcup proh$,

contradicting that the security invariant for refusing has been enforced at time $time - 1$.

Now, regarding lying, the strict alternative given above means that for all $d' \in U$ the uniquely determined harmless version in the set $\{query_{time}, U \backslash query_{time}\}$ is returned, independently of whether it is correct or lied. The same observation applies for the combination. Accordingly, for both approaches we have

$$bestcv_{time} = bestcv_{time-1} \cap U = bestcv_{time-1}\,, \tag{13}$$

together with (12) and the induction assumption implying the claim. □