A Logic for Authorization Provenance

Jinwei Hu[†,‡] Yan Zhang[‡] Ruixuan Li[†] Zhengding Lu[†]
[†]Intelligent and Distributing Computing Laboratory, College of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, China
[‡]Intelligent Systems Laboratory, School of Computing and Mathematics
University of Western Sydney, Sydney, Australia
{jwhu,rxli,zdlu}@hust.edu.cn yan@scm.uws.edu.au

ABSTRACT

In distributed environments, statements from a number of principals, besides the central trusted party, may influence the derivations of authorization decisions. However, existing authorization logics put few emphasis on this set of principals - *authorization provenance*. Reasoning about provenance enables to (1) defend against a class of attacks, (2) understand and analyze authorizations and the status of policy bases, and (3) obtain potentially efficient logging and auditing guided by provenance information. This paper presents the design and applications of a provenance-enabled authorization logic, called DBT. More specifically, we give a sound and complete axiomatic system of DBT. We also examine a class of provenance-aware policy bases and queries. One can syntactically extract provenance information from the structure of these queries if they are evaluated positively in provenance-aware policy bases. Finally, two case studies are presented to demonstrate possible applications of DBT.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection—*Access controls*; F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Modal logic*

General Terms

Security, Authorization, Provenance, Logic

Keywords

Authorization Provenance, Authorization Logic

1. INTRODUCTION

A declarative and expressive language with an unambiguous semantics seems to be a promising approach to authorizations [7]. Based on this observation, major research efforts have applied logics into the design of policy languages to deal with distributed authorizations [1, 5, 9, 13, 16, 18]. These languages enable flexible ways to specify policies, provide efficient mechanisms for evaluating queries, and lay a solid foundation for reasoning about systems' security.

Most of existing authorization logics, however, ignore one important respect of distributed authorizations - *authorization provenance*. Informally, an authorization provenance denotes a set of agents whose statements are referenced in the deduction of an authorization decision. In traditional centralized authorizations, a central trusted party has the complete knowledge of requesting users and requested resources; thus this party can make decisions all by itself. In contrast, no such entity exists in distributed environments, and systems have to employ mechanisms like delegations to facilitate distributed authorizations. Accordingly, besides the central party, a number of agents (e.g., delegatees) play a role together in making access control decisions and are responsible for authorizations.

Suppose that Alice is the warden of a Building Y and that Y requires that if the statement "r_0 : Alice **believes** `opendoor`" can be derived, the request to open door would be allowed. One essential insight is that existing authorization logics do not distinguish between the following two cases where r_0 is reached in different ways.

CASE1	r_1:	Alice **believes** `opendoor`
CASE2	r_2:	Alice **delegates to** Bob `opendoor`
	r_3:	Bob **delegates to** Cathy `opendoor`
	r_4:	Cathy **delegates to** David `opendoor`
	r_5:	David **delegates to** Emma `opendoor`
	r_6:	Emma **believes** `opendoor`

Though r_0 can be concluded in both cases, the reasons why Alice **believes** `opendoor` are different. In CASE1, it is because Alice herself, whereas Bob, Cathy, David, and Emma play a role in CASE2.

There are several compelling reasons why it is advantageous to distinguish the two cases above and reason about authorization provenance. First, host security may be compromised if provenance is not taken into account when making authorization decisions [21, 24]. Wang et al., [24] found that users may abuse delegations to circumvent security policies, if the provenance of users' privileges are not examined. Again, in [21], authors pointed out that, while *Discretionary Access Control* (DAC) models is widely used and deployed in commodity operating systems, they fail to defense against trojan horses; because existing enforcement of DAC cannot correctly identify requests' true origins. It is worth noting that both the reasons and defense mechanisms of these attacks are closely related to authorization provenance.

Second, auditing is an indispensable part of a secure system. One objective of auditing is to identify from where security breaches

ASIACCS'10 April 13–16, 2010, Beijing, China.

started . There arises a trend to include proofs of authorization decisions in system logs for auditing [23]. Armed with the ability to reason about authorization provenance, one may make more use of logs. For example, since provenance record the agents involved, they can help trace back to the origins of security compromises. A log with proofs based on a provenance-enabled logic brings some advantages [15]: (1) one may identify the approximate reasons of an improper authorization before a detailed log analysis; (2) one may pre-classify the log data according to provenance before auditing; the uncategorized log could be of large volume and thus are likely to result in high overhead.

Third, provenance information helps enforcing and analyzing availability and security. Putting restrictions on authorization provenance may prevent insiders' misuse of their privileges. Suppose that the management board of Building Y is composed of Alice, Bob, Cathy, and David and that it is required that whether or not to open door be determined only by the board members. Then, to meet the requirement, each request to open door is asked to be accompanied with a proof of "**due to** Alice, Bob , Cathy, and/or David, Alice **believes** `opendoor`" but not simply that of "Alice **believes** `opendoor`". In CASE2, the door would not be opened for Emma, because Emma is an indispensable agent for the conclusion that "Alice **believes** `opendoor`" to be reached. Hence, the delegation from David to Emma is actually ignored, thus preventing David's misuse and neglect. On the other hand, the query "**due to** Alice, Bob , Cathy, and/or David, Alice **believes** `opendoor`" can also be interpreted as an availability query. Give a policy base, this query is asking whether "Alice **believes** `opendoor`" could be derived by the present statements from Alice, Bob , Cathy, and/or David. If the answer is negative, this implies that these four agents together fail to open door, according to the current policy base.

Finally, since authorization provenance is meta-information about authorizations, explicitly representing provenance help understanding and analyzing policy bases and authorizations. Originally, the need to manage provenance arises from data and scientific applications; and it attracts considerable research efforts in the database and scientific workflow areas [8, 22]. One essential motivation there is that, the provenance of a scientific result or a data computing result is considered as important as the result itself so that one may analyze and evaluate properties of the result such as trustworthiness and completeness. In distributed environments, access control systems have to take the dynamics and flexibility into account and thus lose partial control of who may influence authorizations. Authorization provenance provides an important way to understand how and why authorizations are deduced and to analyze the status of policy bases.

From the above observations, we attempt to develop an authorization logic which treats provenance explicitly. This logic should build upon existing authorization logics, but with more expressive power. One mainstream approach to authorization logics stems from applications of *propositional* modal logic into policy languages, interpreting policies using the Kripke structures [1, 12]. This approach provides a formal platform to study belief, trust, authorization, ect., and their interactions, whereas it lacks in efficient query evaluation algorithms. Fortunately, to enforce policies specified in these logics, proof-carrying authorization seems to be a practical alternative [2, 3, 4, 17], coming to our rescue. In literature, a logic BT (Belief and Trust) [20] is designed to represent belief and trust (delegation) and their relations, which are indispensable in distributed authorizations. Based on BT, we develop a new logic, named DBT (**Due to**, Belief and Trust), which enables reasoning about authorization provenance.

DBT extends the BT logic by introducing a new modal operator D_i for each agent i into the underlying distributed authorizations. $\mathrm{D}_i\varphi$ is designed to express the provenance of φ. While DBT introduces a modal operator for provenance, care is taken to preserve all features of the BT logic for belief and trust. Thus, we integrate the belief, trust, and provenance within a unified logical framework. In summary, our main contributions are as follows.

- We propose a new logic, DBT, which is able to effectively express authorization provenance. We also present a family of axiomatic systems of DBT with increasing reasoning power.
- We study a type of *provenance-aware* policy bases and queries. If a provenance-aware query is evaluated positively against a provenance-aware policy base, one can extract provenance information of the corresponding authorization from the syntactical structure of the query.
- Finally, we give two case studies, which exemplify DBT's motivations and demonstrate possible applications of DBT in complex problem domains.

The rest of the paper is organized as follows. The logic DBT (syntax and semantics) is introduced in Section 2, followed by presentation of the axiomatic systems in Section 3. Section 4 provides an example application of DBT. Provenance-aware policy bases and queries are discussed in Section 5. Section 6 presents an example application of provenance-aware policy bases. Finally, related works and conclusions are addressed in Section 7 and 8, respectively. Due to space limits, we only give proof sketches of propositions and theorems in Appendix.

2. THE ACCESS CONTROL LOGIC DBT

2.1 Syntax

Consider a finite set of agents $\mathcal{AG} = \{1, \cdots, \mathfrak{N}\}$. We have three types of modal operators for each agent i: B_i, T^i_j, and D_i. $\mathrm{B}_i\varphi$ means that agent i *believes* φ or that i *says* φ; and $\mathrm{T}^i_j\varphi$ reads that agent i *trusts* agent j on φ or that i *delegates* φ *to* j. $\mathrm{D}_i\varphi$ means that "*due to* agent i, φ holds" or that i *causes* that φ holds. A subset AE of $\mathcal{AG}$ is called an *agent expression*. Given an $AE \subseteq \mathcal{AG}$, we also define an operator D_{AE} based on D_i for each $i \in AE$. $\mathrm{D}_{AE}\varphi$ means that the set AE of agents together cause φ. Let `Prop` be a set of primitive propositions. The set `WFF` of well-formed formulas (wff) is inductively defined as follows:

$$\begin{aligned}\varphi \quad ::= \quad & p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \Rightarrow \varphi \mid \\ & \mathrm{B}_i\varphi \mid \mathrm{D}_i\varphi \mid \mathrm{D}_{AE}\varphi \mid \mathrm{T}^i_j\varphi\end{aligned}$$

A *policy base* PB is a finite subset of `WFF` and a *query* is a `WFF` formula. We refer to as LOCAL the agent who enforces access control policies. For example, Building Y is LOCAL. LOCAL is the root of trust which protects the requested resources, assembles the policy base, and make access control decisions. When receiving a request to access resources, LOCAL asks for the proof of a corresponding query; we assume that either LOCAL searches for the proof using approaches like those in [4, 3], or LOCAL just verifies a proof submitted by the requester, as in [2]. If the proof is found or verified to be correct by LOCAL, then the request is permitted and otherwise denied. The policy base is composed of the formulas translated from LOCAL's own policies, from credentials that other agents sent to LOCAL, and from the credentials submitted along with the request.

Agents issue credentials to specify their beliefs and delegations. To ensure integrity, credentials are often signed by issuers' private

keys. Credentials are interpreted by DBT formulas. For instance, the credential that "B says that the alarm system in Building Y is in working order" is translated as $\mathrm{B}_{\mathsf{B}}\texttt{alarminorder}$. Suppose that A delegates to B the task of checking if the alarm is in order; we may express this delegation as $\mathrm{T}^{\mathsf{A}}_{\mathsf{B}}\texttt{alarminorder}$. Still, DBT can be more expressive. The formula

$$\mathrm{D}_{\mathsf{B}}\mathrm{B}_{\mathsf{A}}\texttt{alarminorder}$$

represents that due to B, A believes that the alarm is in working order. In this paper, agents could be credential issuers, principals that request accesses, LOCAL, and other entities that may be involved in authorization systems. We may use A, B, C, D, E, and L as a shorthand for the agents Alice, Bob, Cathy, David, Emma, and LOCAL, respectively.

2.2 Semantics

We define a semantics of DBT based on Kripke structures. A *Kripke structure* $\mathcal{M}$ is a tuple $\langle W, \pi, \mathcal{B}_i, \mathcal{D}_i, \mathcal{T}^i_j\rangle$ $(i, j \in \mathcal{AG}; i \neq j)$, where (1) W is a set of states (possible worlds), (2) $\pi : W \mapsto 2^{\texttt{Prop}}$ is a labeling function which maps each state to a subset P of $\texttt{Prop}$ such that, in this state, any $p \in P$ is true and any $p \in \texttt{Prop}\backslash P$ is false, (3) $\mathcal{B}_i \subseteq W \times W$ is a serial, transitive and Euclidean binary relation on W, (4) $\mathcal{D}_i \subseteq W \times W$ is a binary relation on W, and (5) $\mathcal{T}^i_j \subseteq W \times 2^W$ is a binary relation between W and its power set.

Definition 1. ($\models$) Given a structure $\mathcal{M} = \langle W, \pi, \mathcal{B}_i, \mathcal{D}_i, \mathcal{T}^i_j\rangle$, $w \in W$, and a formula φ, let $\mathcal{D}_{AE} = \bigcap_{i\in AE} \mathcal{D}_i$. We define the satisfaction relation $\models$ as follows:

1. $\langle \mathcal{M}, w\rangle \models p$ iff $p \in \pi(w)$,
2. $\langle \mathcal{M}, w\rangle \models \neg\varphi$ iff $\langle \mathcal{M}, w\rangle \not\models \varphi$,
3. $\langle \mathcal{M}, w\rangle \models \varphi_1 \wedge \varphi_2$ iff $\langle \mathcal{M}, w\rangle \models \varphi_1$ and $\langle \mathcal{M}, w\rangle \models \varphi_2$,
4. $\langle \mathcal{M}, w\rangle \models \varphi_1 \Rightarrow \varphi_2$ iff $\langle \mathcal{M}, w\rangle \not\models \varphi_1$, or $\langle \mathcal{M}, w\rangle \models \varphi_2$,
5. $\langle \mathcal{M}, w\rangle \models \mathrm{B}_i\varphi$ iff $\langle \mathcal{M}, v\rangle \models \varphi$ for all v such that $(w, v) \in \mathcal{B}_i$,
6. $\langle \mathcal{M}, w\rangle \models \mathrm{D}_i\varphi$ iff $\langle \mathcal{M}, v\rangle \models \varphi$ for all v such that $(w, v) \in \mathcal{D}_i$,
7. $\langle \mathcal{M}, w\rangle \models \mathrm{D}_{AE}\varphi$ iff $\langle \mathcal{M}, v\rangle \models \varphi$ for all v such that $(w, v) \in \mathcal{D}_{AE}$, and
8. $\langle \mathcal{M}, w\rangle \models \mathrm{T}^i_j\varphi$ iff $(w, \lceil\varphi\rceil) \in \mathcal{T}^i_j$, where $\lceil\varphi\rceil = \{v \in W \mid \langle \mathcal{M}, v\rangle \models \varphi\}$.

While the first four items in the definition of $\models$ are standard [6], some explanations for the last four items seem necessary. From item 5, we can see that B_i is a classical **KD45** belief operator: given that i stays in the state w, i checks the truth of φ in all states v that i believes to be possible.

From the definition of $\mathcal{D}_i$ and item 6, the operator D_i is a classical **K** operator. The intuition of $(w, v) \in \mathcal{D}_i$ (i.e., the state w is $\mathcal{D}_i$-accessible to state v) is that, if agent i stays in the state w, then i could possibly make v a reality (i.e., transform the system in question from the state w to the state v). $\mathrm{D}_i\varphi$ holds in a state w (i.e., $\langle \mathcal{M}, w\rangle \models \mathrm{D}_i\varphi$) if and only if φ is true in all the states v that i could possibly have taken w to. Thus, $\mathrm{D}_i\varphi$ means that i causes φ to hold. Item 7 informs that D_{AE} is an operator for a group of agents: $j \in AE$. $(w, v) \in \mathcal{D}_{AE}$ means that v is a state that every agent in AE can possibly bring about at the state w.

$\mathrm{T}^i_j\varphi$ is designed to mean that i trusts j on φ. As in item 8, the semantics of $\mathrm{T}^i_j\varphi$ is given through a *neighborhood semantics*: if the truth set of φ (i.e., $\lceil\varphi\rceil$) belongs to $\mathcal{T}^i_j(w)$, then $\mathrm{T}^i_j\varphi$ holds at the state w. Among others, an advantage of the neighborhood semantics is that, it is now refutable that $\mathrm{T}^i_j\varphi \Rightarrow \neg\mathrm{T}^i_j\neg\varphi$, which is valid if the semantics of T^i_j is otherwise defined via normal possible worlds semantics. Hence, A is able to specify $\mathrm{T}^{\mathsf{A}}_{\mathsf{B}}\texttt{alarminorder}$ and $\mathrm{T}^{\mathsf{A}}_{\mathsf{B}}\neg\texttt{alarminorder}$ to let B have discretion to judge if the alarm is in working order or not.

AXIOMS
P: all tautologies of the propositional calculus;
B1: $(\mathrm{B}_i\varphi \wedge \mathrm{B}_i(\varphi \Rightarrow \psi)) \Rightarrow \mathrm{B}_i\psi$ B2: $\neg\mathrm{B}_i\bot$
B3: $\mathrm{B}_i\varphi \Rightarrow \mathrm{B}_i\mathrm{B}_i\varphi$ B4: $\neg\mathrm{B}_i\varphi \Rightarrow \mathrm{B}_i\neg\mathrm{B}_i\varphi$
D1: $(\mathrm{D}_i\varphi \wedge \mathrm{D}_i(\varphi \Rightarrow \psi)) \Rightarrow \mathrm{D}_i\psi$
D2: $(\mathrm{D}_{AE}\varphi \wedge \mathrm{D}_{AE}(\varphi \Rightarrow \psi)) \Rightarrow \mathrm{D}_{AE}\psi$
D3: $\mathrm{D}_{AE_1}\varphi \Rightarrow \mathrm{D}_{AE_2}\varphi$, if $AE_1 \subseteq AE_2$
D4: $\mathrm{D}_{AE}\varphi \Leftrightarrow \mathrm{D}_i\varphi$, if $AE = \{i\}$, $i \in \mathcal{AG}$
SBT (Self-Believe Trust): $\mathrm{B}_i\mathrm{T}^i_j\varphi \Leftrightarrow \mathrm{T}^i_j\varphi$
RULES OF INFERENCE
R1 (Modus ponens, MP): from $\vdash \varphi$ and $\vdash \varphi \Rightarrow \psi$ infer $\vdash \psi$
R2 (Generalization, Gen): from $\vdash \varphi$ infer $\vdash \mathrm{B}_i\varphi$ and $\vdash \mathrm{D}_i\varphi$
R3: from $\vdash \varphi \Leftrightarrow \psi$ infer $\vdash \mathrm{T}^i_j\varphi \Leftrightarrow \mathrm{T}^i_j\psi$

Figure 1: The axiomatic system DBT_1

3. THE AXIOMATIC SYSTEMS

We develop a family of axiomatic systems for DBT (i.e., DBT_1, DBT_2, DBT_3, and DBT_4), with increasing reasoning power. This is achieved by imposing a set of constraints on the models. In this section, we introduce these constraints, their intuitions, and the resulting axiomatic systems.

The basic axiom system for DBT_1 is shown in Figure 1. Following [20], DBT_1 includes the axiom SBT, which means that agents should be self aware of and believes their delegations. We believe this is a reasonable assumption. Because credentials are signed and issued by the agents, who should be aware of the credential contents. Rules of inference and axioms except SBT in DBT_1 are standard. Because of the axiom D4, for any $i \in \mathcal{AG}$, we write $\mathrm{D}_{\{i\}}\varphi$ and $\mathrm{D}_i\varphi$ interchangeably in this paper.

Let the set of Kripke models defined in Definition 1 be $\mathbf{MP}_0$ and the models for DBT_1 be $\mathbf{MP}_1$. We define $\mathbf{MP}_1 = \{\mathcal{M} \in \mathbf{MP}_0 \mid \mathcal{M}$ satisfies $C_{\mathrm{SBT}}\}$.

$$C_{\mathbf{SBT}}: \ \mathcal{T}^i_j(w) = \textstyle\bigcap_{u\in\mathcal{B}_i(w)} \mathcal{T}^i_j(u).$$

Example 1. DBT_1 sets a basic stage for reasoning about provenance. Take the Building Y for example. Suppose that we have

$$\mathrm{D}_{\mathsf{A}}\mathrm{B}_{\mathsf{A}}(\texttt{alarmoff} \Rightarrow \texttt{opendoor}) \quad (1)$$

$$\mathrm{D}_{\mathsf{B}}\mathrm{B}_{\mathsf{A}}\texttt{alarmoff} \quad (2)$$

One can derive that $\mathrm{D}_{\{\mathsf{A},\mathsf{B}\}}\mathrm{B}_{\mathsf{A}}\texttt{opendoor}$. That means, because of both A and B, Building Y opens the door.

However, there is no connection among belief, trust, and provenance in DBT_1. Therefore, DBT_1 is of limited usefulness as for authorization provenance. To capture authorization provenance, especially those in distributed environments, we introduce some other inference laws.

Agents may still reason about their own beliefs and trusts, even in distributed settings. An agent can make access control decisions based on its own knowledge. Since the agent makes no use of others' statements (or credentials), those decisions are due to itself.

Hence the following two axioms are desired:

$$\text{SRB (Self-Responsible Belief):} \vdash \mathrm{D}_i\mathrm{B}_i\varphi \Leftrightarrow \mathrm{B}_i\varphi.$$
$$\text{SRT (Self-Responsible Trust):} \vdash \mathrm{D}_i\mathrm{T}^i_j\varphi \Leftrightarrow \mathrm{T}^i_j\varphi.$$

SRB and SRT require that agents be self-responsible for their beliefs and delegations (trusts), respectively. In other words, the provenance of i's conclusions, whose deductions are independent of all agents except i, is i itself. We refer to the logic as $DBT_2 = DBT_1 + \text{SRB} + \text{SRT}$.

Example 2. Suppose that

$$\mathrm{B}_{\mathsf{A}}(\texttt{alarmoff} \Rightarrow \texttt{opendoor})$$
$$\mathrm{B}_{\mathsf{A}}\texttt{alarmoff}$$

According to DBT_2, they imply that $\mathrm{B}_{\mathsf{A}}\texttt{opendoor}$. That is, A believes opendoor. Afterwards, if one wants to check who is responsible for this authorization (e.g., for auditing), one can derive that $\mathrm{D}_{\mathsf{A}}\mathrm{B}_{\mathsf{A}}\texttt{opendoor}$ but not that $\mathrm{D}_{\mathsf{X}}\mathrm{B}_{\mathsf{A}}\texttt{opendoor}$, where X is an agent but $\mathsf{X} \neq \mathsf{A}$. Hence, that authorization is granted simply due to A itself but nobody else.

Let $\mathbf{MP}_2$ denote the models for DBT_2. Then $\mathbf{MP}_2 = \{\mathcal{M} \in \mathbf{MP}_1 \mid \mathcal{M} \text{ satisfies } C_{\text{SRB}} \text{ and } C_{\text{SRT}}\}$.

$C_{\textbf{SRB}}$: $\mathcal{B}_i(w) = \mathcal{D}_i \circ \mathcal{B}_i(w)$.[1]

$C_{\textbf{SRT}}$: $\mathcal{T}^i_j(w) = \bigcap_{u \in \mathcal{D}_i(w)} \mathcal{T}^i_j(u)$.

Nevertheless, agents may partially depend on delegations to make decisions in distributed authorization. Take Example 2 for instance. A is not able to check alarmoff, but A trusts B on this matter. Then, the door of Building Y may be opened if B presents a credential, showing that the alarm is set off. Therefore, we need to capture the effects of delegations. To this end, the following axioms seem plausible.

$$\text{Dlgt (Delegation):} \vdash \mathrm{T}^i_j\varphi \wedge \mathrm{B}_j\varphi \Rightarrow \mathrm{D}_j\mathrm{B}_i\varphi$$
$$\text{i-centric-Dlgt:} \vdash \mathrm{T}^i_j\varphi \wedge \mathrm{T}^j_k\varphi \Rightarrow \mathrm{D}_j\mathrm{T}^i_k\varphi$$

Dlgt and i-centric-Dlgt make a connection among belief operator B_i, trust operator T^i_j, and provenance operator D_j. Dlgt means that, if agent i places trust on j about φ and j believes φ , then agent j causes i to believe φ. i-centric-Dlgt chains the delegations; that is, if as for φ, agent i trusts j, who in turn trusts k, then it holds that j causes i to trust k on φ. By these two axioms, the trusted agent is recorded as provenance of the derived conclusion. We refer to the logic as $DBT_3 = DBT_2 + \text{Dlgt} + \text{i-centric-Dlgt}$.

Example 3. Example 2 continued. A believes that if the alarm is off then the door could be opened and A also delegates to B the judgement of alarmoff. B presents a credential to confirm that the alarm is off.

$$\mathrm{B}_{\mathsf{A}}(\texttt{alarmoff} \Rightarrow \texttt{opendoor}) \quad (3)$$
$$\mathrm{T}^{\mathsf{A}}_{\mathsf{B}}\texttt{alarmoff} \quad (4)$$
$$\mathrm{B}_{\mathsf{B}}\texttt{alarmoff} \quad (5)$$

Then by the axiom Dlgt, from (4) and (5), it follows that (6). And by the axioms D4 and D3, one can derive (7) and (8), from (6) and (3), respectively. Finally, by applying the axioms of D2 and B1 to (7) and (8), we have (9).

$$\mathrm{D}_{\mathsf{B}}\mathrm{B}_{\mathsf{A}}\texttt{alarmoff} \quad (6)$$
$$\mathrm{D}_{\{\mathsf{A},\mathsf{B}\}}\mathrm{B}_{\mathsf{A}}\texttt{alarmoff} \quad (7)$$
$$\mathrm{D}_{\{\mathsf{A},\mathsf{B}\}}\mathrm{B}_{\mathsf{A}}(\texttt{alarmoff} \Rightarrow \texttt{opendoor}) \quad (8)$$
$$\mathrm{D}_{\{\mathsf{A},\mathsf{B}\}}\mathrm{B}_{\mathsf{A}}\texttt{opendoor} \quad (9)$$

Suppose that Building Y configures the door to open if a proof for $\mathrm{D}_{\{\mathsf{A},\mathsf{B}\}}\mathrm{B}_{\mathsf{A}}\texttt{opendoor}$ is found or verified to be correct. Then by the above reasoning, the door should open. Further, Y could demand that, "as long as $\mathrm{B}_{\mathsf{A}}\texttt{opendoor}$ with the help of A herself, B, C, and D (i.e., the board members), the door should open" by asking for a proof of $\mathrm{D}_{\{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\}}\mathrm{B}_{\mathsf{A}}\texttt{opendoor}$, which can be obtained by applying the axiom D3 to (9). Taken extremely, Y may ask for a proof of $\mathrm{D}_{\mathcal{AG}}\mathrm{B}_{\mathsf{A}}\texttt{opendoor}$ if Y does not care about authorization provenance (i.e., no matter how A's belief in opendoor is gained). Recall that $\mathcal{AG}$ is the set of all agents involved.

Suppose that B himself could not check the alarm either and just delegates this task to C, as represented by (10), and that (5) no longer holds, whereas (11) is true instead.

$$\mathrm{T}^{\mathsf{B}}_{\mathsf{C}}\texttt{alarmoff} \quad (10)$$
$$\mathrm{B}_{\mathsf{C}}\texttt{alarmoff} \quad (11)$$

By the axiom i-centric-Dlgt, from (4) and (10), it follows that (12).

$$\mathrm{D}_{\mathsf{B}}\mathrm{T}^{\mathsf{A}}_{\mathsf{C}}\texttt{alarmoff} \quad (12)$$
$$\mathrm{D}_{\{\mathsf{B},\mathsf{C}\}}\mathrm{B}_{\mathsf{A}}\texttt{alarmoff} \quad (13)$$
$$\mathrm{D}_{\{\mathsf{A},\mathsf{B},\mathsf{C}\}}\mathrm{B}_{\mathsf{A}}\texttt{opendoor} \quad (14)$$

By applying the axioms of SRB, D4, D3, Dlgt, and AE-Red (introduced below), from (12) and (11) we obtain (13), which, together with (3), derives (14). The derivations of (13) and (14) depend on an axiom AE-Red. AE-Red is not included in DBT_3. but in DBT_4, which will be introduced below.

By imposing the following constraints C_{Dlgt} and $C_{\text{i-centric-Dlgt}}$ on $\mathbf{MP}_2$, we obtain the models for DBT_3, denoted as $\mathbf{MP}_3$. That is, $\mathbf{MP}_3 = \{\mathcal{M} \in \mathbf{MP}_2 \mid \mathcal{M} \text{ satisfies } C_{\text{Dlgt}} \text{ and } C_{\text{i-centric-Dlgt}}\}$.

$C_{\textbf{Dlgt}}$: for all $S \in \mathcal{T}^i_j(w)$, if $\mathcal{B}_j(w) \subseteq S$, then $\mathcal{D}_j \circ \mathcal{B}_i(w) \subseteq S$.

$C_{\textbf{i-centric-Dlgt}}$: $\mathcal{T}^i_j(w) \cap \mathcal{T}^j_k(w) \subseteq \bigcap_{u \in \mathcal{D}_j(w)} \mathcal{T}^i_k(u)$.

With DBT_3, provenance is recorded in the derivations of authorizations. To facilitate explorations of authorization provenance, they are expected to meet some requirements. Here we are interested in two minimal ones.

$$\text{Red:} \vdash \mathrm{D}_i\mathrm{D}_i\varphi \Rightarrow \mathrm{D}_i\varphi$$
$$\text{AE-Red:} \vdash \mathrm{D}_{AE}\mathrm{D}_{AE}\varphi \Rightarrow \mathrm{D}_{AE}\varphi$$

With Red and AE-Red, the provenance of any formula φ needs not to be repeated: if it is because of i itself that i causes φ holds, then it follows that it is just because of i that φ holds; and this is also the case for a group of agents AE. We refer to the logic as $DBT_4 = DBT_3 + \text{Red} + \text{AE-Red}$.

Example 4. An instance of the axiom AE-Red is already used in Example 3. Here we describe the derivation of the formula (13) in Example 3. By applying the axioms of D4 and D3 to (12), we have (15). By applying the axioms of SRB, D4 and D3 to (11), we have (16). And then applications of D2 and Dlgt to (15) and (16) give rise to (17), from which one can derive (18) with the axioms

[1] Suppose that $R \subseteq \mathcal{X} \times \mathcal{Y}$ is a binary relation between $\mathcal{X}$ and $\mathcal{Y}$. Let $R(x)$ be the set $\{y \in \mathcal{Y} \mid (x, y) \in R\}$. Assuming $Q \subseteq \mathcal{Y} \times \mathcal{Z}$, let $R \circ Q$ be a binary relation between $\mathcal{X}$ and $\mathcal{Z}$ such that $R \circ Q = \{(x, z) \mid \exists y \in \mathcal{Y} : \ y \in R(x) \wedge z \in Q(y)\}$.

D4 and D3. Finally, by applying an instance of AE-Red to (18), we have (13).

$$\mathrm{D}_{\{\mathsf{B},\mathsf{C}\}}\mathrm{T}^{\mathsf{A}}_{\mathsf{C}}\mathtt{alarmoff} \quad (15)$$

$$\mathrm{D}_{\{\mathsf{B},\mathsf{C}\}}\mathrm{B}_{\mathsf{C}}\mathtt{alarmoff} \quad (16)$$

$$\mathrm{D}_{\{\mathsf{B},\mathsf{C}\}}\mathrm{D}_{\mathsf{C}}\mathrm{B}_{\mathsf{A}}\mathtt{alarmoff} \quad (17)$$

$$\mathrm{D}_{\{\mathsf{B},\mathsf{C}\}}\mathrm{D}_{\{\mathsf{B},\mathsf{C}\}}\mathrm{B}_{\mathsf{C}}\mathtt{alarmoff} \quad (18)$$

The models of DBT_4, called $\mathbf{MP}_4$, are a subset of $\mathbf{MP}_3$ which satisfies the constrains C_{Red} and $C_{\text{AE-Red}}$. Namely, $\mathbf{MP}_4 = \{\mathcal{M} \in \mathbf{MP}_3 \mid \mathcal{M}$ satisfies C_{Red} and $C_{\text{AE-Red}}\}$.

$C_{\textbf{Red}}$: $\mathcal{D}_i(w) \subseteq \mathcal{D}_i \circ \mathcal{D}_i(w)$.

$C_{\textbf{AE-Red}}$: $\mathcal{D}_{AE}(w) \subseteq \mathcal{D}_{AE} \circ \mathcal{D}_{AE}(w)$.

Given a Kripke structure $\mathcal{M}$, we say that a formula φ is *valid* in $\mathcal{M}$, denoted by $\mathcal{M} \models \varphi$, if $\langle \mathcal{M}, w\rangle \models \varphi$ for all $w \in W$; and for each $1 \leq t \leq 4$, say φ is *valid in* $\boldsymbol{MP}_t$, written $\mathbf{MP}_t \models \varphi$, if φ is valid in all $\mathbf{MP}_t$ models. Also, for each $1 \leq t \leq 4$, we say that a formula φ is *provable* in DBT_t, denoted by $DBT_t \vdash \varphi$, if and only if, there is a finite sequence of $\chi_1, \cdots, \chi_n, \chi_{n+1}$, such that $\varphi = \chi_{n+1}$, and every χ_l is an instance of an axiom in DBT_t or obtainable from an application of an inference rule in DBT_t to $\chi_{l_1} \cdots \chi_{l_m}$, where $l_1, \cdots, l_m < l$. Recall that a policy base is a finite subset of WFF and that a query is a WFF formula. We say that a policy base PB *entails* a query q wrt $\mathbf{MP}_t$, written PB$\models_{\mathbf{MP}_t} q$, if and only if, for all $\mathcal{M} \in \mathbf{MP}_t$ and states w in $\mathcal{M}$, if for all $\psi \in$ PB $\langle \mathcal{M}, w\rangle \models \psi$ then $\langle \mathcal{M}, w\rangle \models q$.

THEOREM 1. *For each $1 \leq t \leq 4$, the axiomatic system DBT_t is sound and complete with respect to $\boldsymbol{MP}_t$.*

COROLLARY 2. *Given a policy base PB and a query q, for each $1 \leq t \leq 4$, PB$\models_{\boldsymbol{MP}_t} q$ if and only if $DBT_t \vdash (\bigwedge PB) \Rightarrow q$.*

4. APPLICATION: DEFENDING AGAINST TROJAN HORSES

4.1 Background

Discretionary Access Control (DAC) is widely supported in modern operating systems to protect systems' resources. However, DAC is vulnerable to trojan horses [21]. A trojan horse is a piece of malicious software which may perform malicious actions. A trojan-exploiting attacker can obtain accesses to resources that otherwise she/he is not authorized to.

Take the scenario in Figure 2 for instance. Following denotations in [21], we express the DAC policies as $wpc(\mathsf{obj}) = \{\mathsf{B}, \mathsf{C}, \mathsf{E}\}$ and $rpc(\mathsf{psw}) = \{\mathsf{C}, \mathsf{D}, \mathsf{E}\}$.[2] B is a malicious agent in terms of reading psw, which he is not authorized to. However, B writes a piece of malicious code into the obj. When proc reads from obj, the malicious code may be planted into proc and executed; then B may take over proc as a result. Afterwards, B could perform unauthorized accesses (e.g., to psw) via proc.

In [21], Mao et al., pointed out that the reason why DAC fails to defend against trojan horses is that, existing enforcement of DAC models cannot correctly identify the true origins of requests. In the example above, when proc issues a request to read psw, the existing enforcements deem that this request is simply from C, thus permitting proc to read psw. Unfortunately, since the malicious code take effects, this request is due to B, who hides behind the proc. Even though B is not allowed to read psw by the DAC policies, he can actually read psw via proc.

[2] wpc is short for *write protection class* and rpc for *read protection class.* Details are referred to [21].

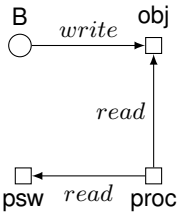

Figure 2: A simple case where a trojan horse is planted. proc is a process run by C, obj is an object (e.g., a file) owned by C, and psw is a sensitive file owned by C. Relevant DAC policies are as follows: C allows B, C and E to write to obj, and C, D, and E to read psw.

Mao et al., proposed a model based on a notion of a *contamination source*, which is the set of all agents. Roughly speaking, the main idea is to associate with each object and process a set of agents, which have ever interacted with it (e.g., read, write, or create it); and deny the request if any member of the contamination source of the request is not allowed to access. We call this Contamination Source based Enforcement *CSE* [21].

Given an object or process r, let $cs(r)$ be r's contamination source. For example, initially since C owns obj and C runs proc, $cs(\mathsf{obj}) = \{\mathsf{C}\}$ and $cs(\mathsf{proc}) = \{\mathsf{C}\}$. After B writes obj, $cs(\mathsf{obj}) = \{\mathsf{B}, \mathsf{C}\}$. And after proc reads obj, $cs(\mathsf{proc}) = cs(\mathsf{obj}) \cup cs(\mathsf{proc}) = \{\mathsf{B}, \mathsf{C}\}$; thus $\mathsf{B} \in cs(\mathsf{proc})$. CSE considers every request issued by proc to be possibly from any member of $cs(\mathsf{proc})$; hence, proc's request to read psw may originate from B. According to the DAC policies, B is not allowed to read psw and thus CSE would deny this request. Therefore CSE forbids B from reading psw, even though B took over proc through the trojan horse planted via obj.

4.2 Formalization of CSE

Using DBT, we may formalize the requesting and decision-making parts of CSE in two steps.

Express the requests associated with contamination sources. Given a process p with $cs(p)$, we represent the request to perform an action a on an object o as

$$\mathrm{D}_{cs(p)}\mathtt{req}(p, a, o).$$

For example, when proc asks for accesses to read psw, the request is specified as $\mathrm{D}_{\{\mathsf{B},\mathsf{C}\}}\mathtt{req}(\mathsf{proc}, read, \mathsf{psw})$.

Express the DAC policies. Assume that, given an object o and its owner u, u designates a set of users $\mathsf{UserSet} = \{\mathsf{u}_1, \cdots, \mathsf{u}_n\}$ who can perform an action a on o.[3] In DBT, this is expressed as

$$\mathrm{D}_{\mathsf{UserSet}}\mathtt{req}(p, a, o) \Rightarrow \mathtt{grant}(p, a, o).$$

For instance, the DAC policy $rpc(\mathsf{psw}) = \{\mathsf{C}, \mathsf{D}, \mathsf{E}\}$ can be represented as $\mathrm{D}_{\{\mathsf{C},\mathsf{D},\mathsf{E}\}}\mathtt{req}(\mathsf{proc}, read, \mathsf{psw}) \Rightarrow \mathtt{grant}(\mathsf{proc}, read, \mathsf{psw})$.

Finally, each time a process p requests to perform an action a on an object o, either p is asked to prove $\mathtt{grant}(p, a, o)$ or L must find a proof for $\mathtt{grant}(p, a, o)$. For instance, when proc requests to read psw, $\mathtt{grant}(\mathsf{proc}, read, \mathsf{psw})$ needs to be proved.

Hence, when B tries to read psw in the guise of proc, the request would be declined. In contrast, if it was D who had writ-

[3] According to [21], there may be exceptions; we can also represent these exceptions in DBT.

ten obj and requests to read psw via proc, the access would be granted. Because D is allowed to read psw by the DAC policies in the first place; the request would be expressed as $\mathrm{D}_{\{\mathsf{C},\mathsf{D}\}}\mathtt{req}(\mathsf{proc}, read, \mathsf{psw})$. And by an instance of the axiom "D3":

$$\mathrm{D}_{\{\mathsf{C},\mathsf{D}\}}\mathtt{req}(\mathsf{proc}, read, \mathsf{psw}) \Rightarrow \mathrm{D}_{\{\mathsf{C},\mathsf{D},\mathsf{E}\}}\mathtt{req}(\mathsf{proc}, read, \mathsf{psw}),$$

it can be concluded that $\mathtt{grant}(\mathsf{proc}, read, \mathsf{psw})$.

Actually, applications of instances of the axiom D3 play an important role in the DBT's formalization of CSE. Supposing that S is the origins of a request to perform a on o, the request is represented as $\mathrm{D}_{\mathsf{S}}\mathtt{req}(p, a, o)$. Since $\mathsf{S} \subseteq \mathsf{UserSet}$ and from the instance of the axiom D3: $\mathrm{D}_{\mathsf{S}}\mathtt{req}(p, a, o) \Rightarrow \mathrm{D}_{\mathsf{UserSet}}\mathtt{req}(p, a, o)$, the access would be granted. However, for any set $\mathsf{S}' \not\subseteq \mathsf{UserSet}$ of users who issue such a request, it would be declined.

The point of this formalization in DBT is to set a stage for CSE techniques to interact with systems, where authorization logics are employed to specify their policy bases. These systems also suffer from trojan horses, in spite of the correct enforcement of their policy bases. This is partially because they neglect the provenance information of requests and thus that of authorizations. Given policy bases formalized in DBT, we may combine CSE techniques in the hope that they can also share the merits resulting from the notion of contamination source.

4.3 Extension

Attack. When it comes to distributed authorizations, CSE might fail to prohibit trojan horses in presence of delegations. We extend the scenario in Figure 2. As shown in Figure 3, this time B changes attacking strategy. Somehow B worms into the favor of E, who (innocently) delegates the action $\mathtt{write}(\mathsf{obj}, input)$ to B.[4] Unfortunately, B's plot would prevail again despite CSE's being in position.

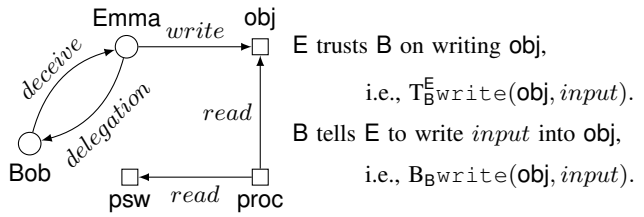

Figure 3: A trojan horse is planted with the help of delegations.

Let us explain B's attack in some details. Recall that $cs(\mathsf{obj}) = \{\mathsf{C}\}$ and $cs(\mathsf{proc}) = \{\mathsf{C}\}$ originally. Since E trusts B on writing $input$ to obj and B tells E to do so, L at E's side can find or correctly verify a proof of E believing writing $input$ to obj and thus allows this action. Note that here the provenance of how E obtains this belief is not counted. Then after obj is written with $input$, $cs(\mathsf{obj}) = \{\mathsf{C}, \mathsf{E}\}$ in the light of CSE.[5] Once proc reads obj, $cs(\mathsf{proc}) = \{\mathsf{C}, \mathsf{E}\}$ according to CSE and the malicious code injected in obj may be executed and B can take over proc as a result. When proc, as a puppet in B's hand, requests to read psw, CSE would permit this action because all members of $cs(\mathsf{proc})$ are allowed to read psw.

We may augment CSE with provenance information to deal with this kind of delegation-exploiting trojan horses. We refer to this extended CSE as *ExCSE*. We first assume that, if actions are performed because of conclusions derived from agents' policy bases, L is able to syntactically check these conclusions. For example, from $\mathrm{D}_{\mathsf{B}}\mathrm{B}_{\mathsf{E}}\mathtt{write}(\mathsf{obj}, input)$, L can tell the information "due to B, $\mathrm{B}_{\mathsf{E}}\mathtt{write}(\mathsf{obj}, input)$ holds". The conclusions should take the form of $\mathrm{D}_{AE}\mathrm{B}_i\varphi$. When $AE = \{i\}$ this means the conclusion is the intention of i itself; otherwise it relies on delegations. The idea is to regard AE as contamination source as well.

We show how ExCSE works by E's example. Suppose that L asks for a proof of $\mathrm{D}_{AE}\mathrm{B}_{\mathsf{E}}\mathtt{write}(\mathsf{obj}, input)$ for $input$ to be written to obj. After L verifies the proof, the contamination source of obj changes as $cs(\mathsf{obj}) = AE \cup cs(\mathsf{obj})$. Since $AE = \{\mathsf{B}\}$, $cs(\mathsf{obj}) = \{\mathsf{B}, \mathsf{C}, \mathsf{E}\}$ after $input$ is written into obj, and $cs(\mathsf{proc}) = \{\mathsf{B}, \mathsf{C}, \mathsf{E}\}$ after proc reads obj. Hence, proc's request to read psw is considered to be possibly from B. And the request can only be formulated as $\mathrm{D}_{\{\mathsf{B},\mathsf{C},\mathsf{E}\}}\mathtt{req}(\mathsf{proc}, read, \mathsf{psw})$, thus would be denied.

Note that ExCSE depends on CSE to track contamination sources. The extension lies in that ExCSE uses DBT to record provenance information when facing with delegations and treat provenance as contamination source. It would be awkward for DBT to track contamination sources after, for example, processes reads objects. However, DBT helps generalize CSE techniques to cope with trojan horses at access control level. In summary, DBT naturally abstracts requests so that CSE could be integrated with logic-based policy bases. On the other hand, with the help of DBT, ExCSE blocks another kind of trojan horse attacks in the context of distributed authorizations.

5. PROVENANCE-AWARE POLICY BASES

Provenance is an important means to understand the delegations present in policy bases. However, how fine-grained provenance information is encoded depends on how policies are authored, even though DBT is able to express provenance. Here we study a class of provenance-aware policy bases (PaPB) and queries. With PaPB, one can syntactically extract the provenance information from provenance-aware queries, if they are evaluated positively against PaPB.

Given $p \in \mathtt{Prop}$, the set $\mathtt{WFF}_{pa}$ of formulas is given by the rule

$$\phi \quad ::= \quad \mathrm{B}_i p \mid \mathrm{T}^i_j p$$

We define a mapping $\mathsf{issuer} : \mathtt{WFF}_{pa} \mapsto \mathcal{AG}$. Given $\phi \in \mathtt{WFF}_{pa}$, $\mathsf{issuer}[\phi] = i$ if $\phi = \mathrm{B}_i p$ or $\phi = \mathrm{T}^i_j p$. $\mathsf{issuer}[\phi]$ denotes the agent who issues the credential corresponding to ϕ.

Definition 2. Given a PB $\subset \mathtt{WFF}_{pa}$, we may construct the *delegation structure* in PB, denoted as $\mathsf{DS}\langle\mathrm{PB}\rangle$. $\mathsf{DS}\langle\mathrm{PB}\rangle$ is a tuple $\langle N, E, \varrho\rangle$, where $N = \bigcup_{\phi\in\mathrm{PB}}\{\mathsf{issuer}[\phi]\}$, $\langle \mathsf{n}_1, \mathsf{n}_2\rangle \in E$ if and only if $\mathrm{T}^{\mathsf{n}_1}_{\mathsf{n}_2} p \in \mathrm{PB}$ for some $p \in \mathtt{Prop}$, and $\varrho : E \mapsto 2^{\mathtt{Prop}}$ is a mapping defined as $p \in \varrho[\langle \mathsf{n}_1, \mathsf{n}_2\rangle]$ if and only if $\mathrm{T}^{\mathsf{n}_1}_{\mathsf{n}_2} p \in \mathrm{PB}$.

Given $\{e_0 = \langle \mathsf{n}, \mathsf{n}_1\rangle, e_1 = \langle \mathsf{n}_1, \mathsf{n}_2\rangle, \cdots, e_l = \langle \mathsf{n}_l, \mathsf{n}'\rangle\} \subseteq E$, we say $\{e_0, \cdots, e_l\}$ is *a path (from n to n') associated with p* if $p \in \varrho[e_0] \cap \cdots \cap \varrho[e_l]$, and write $\langle \mathsf{n}, \mathsf{n}_1, \cdots, \mathsf{n}_l, \mathsf{n}'\rangle$ for short. We say a delegation structure $\mathsf{DS}\langle\cdot\rangle = \langle N, E, \varrho\rangle$ is *acyclic* if there is no path from n to n associated with p for any $\mathsf{n} \in N$ and $p \in \mathtt{Prop}$.

We say PB is a *Provenance aware Policy Base*, denoted as PaPB, if $\mathrm{PB} \subset \mathtt{WFF}_{pa}$ and $\mathsf{DS}\langle\mathrm{PB}\rangle$ is acyclic. The requirement that

[4]For example, B may offer to help E diagnose and configure her computer. Consider a remote assistance scenario: http://technet.microsoft.com/en-us/library/bb457004.aspx.

[5]Here, E finishes this action through some process pr; to simplify explanation, we assume pr's contamination source does not make any difference in granting read-access to psw, or we can simply assume that $cs(pr) = \{\mathsf{E}\}$.

$\mathsf{DS}\langle\text{PB}\rangle$ be acyclic is not ambitious. Because a path from n to n means delegations both originates from and ends at n, which implies that some delegations are not necessary or make no sense.

Definition 3. (Types of queries) Given a query $q = \mathrm{D}_{AE_n} \cdots \mathrm{D}_{AE_1}\phi$, we define the following types of queries:

1. q is an *PaPB-query*, if $\phi \in \mathtt{WFF}_{pa}$,
2. q is a *prov-query*, if q is a PaPB-query and $\mathsf{issuer}[\phi] \notin \bigcup_{1\le l\le n} AE_l$,
3. q is a *prov-aware query*, if q is a prov-query and AE_1 is a singleton set (i.e., $AE_1 = \{j_1\} \subseteq \mathcal{AG}$) and for $2 \le l \le n$, $AE_l = AE_{l-1} \cup \{j_l\}$ where $j_l \in \mathcal{AG}$ and $j_l \notin AE_{l-1}$.

Example 5. Let the proposition p denoting `opendoor`, consider the following queries.

$$\begin{array}{ll} q_1 : \mathrm{D}_{\mathsf{D}}\mathrm{T}^{\mathsf{B}}_{\mathsf{C}}\mathrm{B}_{\mathsf{A}}p & q_4 : \mathrm{D}_{\{\mathsf{B},\mathsf{C},\mathsf{D}\}}\mathrm{B}_{\mathsf{A}}p \\ q_2 : \mathrm{D}_{\{\mathsf{A},\mathsf{B}\}}\mathrm{T}^{\mathsf{A}}_{\mathsf{C}}p & q_5 : \mathrm{D}_{\mathsf{B}}\mathrm{D}_{\{\mathsf{B},\mathsf{C}\}}\mathrm{B}_{\mathsf{A}}p \\ q_3 : \mathrm{D}_{\{\mathsf{B},\mathsf{C}\}}\mathrm{T}^{\mathsf{A}}_{\mathsf{D}}p & q_6 : \mathrm{D}_{\{\mathsf{B},\mathsf{C},\mathsf{D}\}}\mathrm{D}_{\{\mathsf{C},\mathsf{D}\}}\mathrm{D}_{\mathsf{D}}\mathrm{B}_{\mathsf{A}}p \end{array}$$

The query q_1 is not a PaPB-query since $\mathrm{T}^{\mathsf{B}}_{\mathsf{C}}\mathrm{B}_{\mathsf{D}}p \notin \mathtt{WFF}_{pa}$, The query q_2 is a PaPB-query, but not a prov-query because $\mathsf{issuer}[\mathrm{T}^{\mathsf{A}}_{\mathsf{C}}p] \in \{\mathsf{A},\mathsf{B}\}$, whereas q_3 is a prov-query. Queries q_4 and q_5 are also prov-queries, but not prov-aware queries. Finally, q_6 is a prov-aware query.

We now present some characteristic properties of these queries.

Given a PaPB-query $q = \mathrm{D}_{AE_n} \cdots \mathrm{D}_{AE_1}\phi$, we define that, for any $2 \le l \le n$, $\mathsf{index}[AE_l, q] = 1 + \mathsf{index}[AE_{l-1}, q]$ and $\mathsf{index}[AE_1, q] = 1$. Supposing that q is a prov-aware query, then $\mathsf{index}[AE_l, q] = |AE_l|$ for $1 \le l \le n$, where $|AE_l|$ denotes the cardinality of the set AE_l. Denote the path $\langle \mathsf{n}_1, \mathsf{n}_2, \cdots, \mathsf{n}_{l-1}, \mathsf{n}_l\rangle$ as $\mathsf{path}[\mathsf{n}_1, \mathsf{n}_l]$ and the set $\{\mathsf{n}_1, \cdots, \mathsf{n}_{l-1}\}$ as $\mathsf{ag@path}[\mathsf{n}_1, \mathsf{n}_l]$. Note that $\mathsf{n}_l \notin \mathsf{ag@path}[\mathsf{n}_1, \mathsf{n}_l]$.

PROPOSITION 3. *Given PaPB and a PaPB-query $q = D_{AE_n} \cdots D_{AE_1}\phi$, if PaPB$\models_{\mathbf{MP}_4} q$, then*

- *when ϕ is of the form $T^i_j p$, $\mathsf{DS}\langle$PaPB$\rangle$ has a path $\mathsf{path}[i, j]$ from i to j associated with p satisfying the following claims;*
- *when ϕ is of the form $B_i p$, if there exists some $1 \le l \le n$ such that $i \notin AE_l$ and $i \in AE_t$ holds for all $1 \le t < l$. then there exists an agent $j \in AE_l$ such that $\mathsf{DS}\langle$PaPB$\rangle$ has a path $\mathsf{path}[i, j]$ from i to j associated with p satisfying the following claims.*

CLAIM1 *$\mathsf{ag@path}[i, j] \subseteq \bigcup_{1\le t\le n} AE_t \cup \{i\}$.*

CLAIM2 *PaPB$\models_{\mathbf{MP}_4} D_{AE'_n} \cdots D_{AE'_1}\phi$, where for $1 \le t \le n$, $AE'_t = AE_t \cap (\mathsf{ag@path}[i, j] \cup \{j\})$.*

COROLLARY 4. *Given a PaPB-query $q = D_{AE}\phi$ where $\mathsf{index}[AE, q] = 1$, if ϕ is of the form $T^i_j p$ then $\mathsf{DS}\langle$PaPB$\rangle$ has a path $\mathsf{path}[i, j]$ such that $\mathsf{ag@path}[i, j] \subseteq AE \cup \{i\}$ and that PaPB$\models_{\mathbf{MP}_4} D_{AE'}\phi$ where $AE' = AE \cap (\mathsf{ag@path}[i, j]\backslash\{i\})$; and if ϕ is of the form $B_i p$ and $i \notin AE$ then there exists an agent $j \in AE$ such that $\mathsf{DS}\langle$PaPB$\rangle$ has a path $\mathsf{path}[i, j]$ such that $\mathsf{ag@path}[i, j] \subseteq AE \cup \{i\}$ and that PaPB$\models_{\mathbf{MP}_4} D_{AE'}\phi$ where $AE' = AE \cap (\mathsf{ag@path}[i, j]\backslash\{i\})$.*

Example 6. Consider the query $q_4 = \mathrm{D}_{\{\mathsf{B},\mathsf{C},\mathsf{D}\}}\mathrm{B}_{\mathsf{A}}p$ in Example 5. If PaPB$\models_{\mathbf{MP}_4} q_4$, then for certain agent $\mathsf{X} \in \{\mathsf{B},\mathsf{C},\mathsf{D}\}$, $\mathsf{DS}\langle$PaPB$\rangle$ has a path $\mathsf{path}[\mathsf{A},\mathsf{X}]$ such that $\mathsf{ag@path}[\mathsf{A},\mathsf{X}] \subseteq \{\mathsf{B},\mathsf{C},\mathsf{D}\} \cup \{\mathsf{A}\}$. That is, some agents in the set $\{\mathsf{B},\mathsf{C},\mathsf{D}\}$ made A believe p. If agents $\{\mathsf{B},\mathsf{C},\mathsf{D}\}$ should have this capability, then q_4 is asking about availability and a positive answer is expected, (i.e., some agents in the set $\{\mathsf{B},\mathsf{C},\mathsf{D}\}$ can make A believe p). If not, then q_4 is asking about security and a negative answer is expected.

q_4 serves the purpose of the query "**due to** A, B , C, and/or D, A **believes** `opendoor`" in Section 1. Suppose that PaPB$\models_{\mathbf{MP}_4} q_4$, we have PaPB$\models_{\mathbf{MP}_4} \mathrm{D}_{\{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\}}\mathrm{B}_{\mathsf{A}}p$ by the axiom D3. On the other hand, assume that PaPB$\models_{\mathbf{MP}_4} \mathrm{D}_{\{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\}}\mathrm{B}_{\mathsf{A}}p$. From Corollary 4, we have PaPB$\models_{\mathbf{MP}_4} q_4$. Hence PaPB$\models_{\mathbf{MP}_4} q_4$ if and only if PaPB$\models_{\mathbf{MP}_4} \mathrm{D}_{\{\mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\}}\mathrm{B}_{\mathsf{A}}p$.

Consider another query $q_7 = \mathrm{D}_{\{\mathsf{B},\mathsf{C},\mathsf{D}\}}\mathrm{D}_{\mathsf{C}}\mathrm{B}_{\mathsf{A}}p$. If PaPB$\models_{\mathbf{MP}_4} q_7$, then $\mathsf{DS}\langle$PaPB$\rangle$ has a path $\mathsf{path}[\mathsf{A},\mathsf{C}]$ such that $\mathsf{ag@path}[\mathsf{A},\mathsf{C}] \subseteq \{\mathsf{B},\mathsf{C},\mathsf{D}\} \cup \{\mathsf{C}\} \cup \{\mathsf{A}\}$. Suppose that, actually, $\mathsf{path}[\mathsf{A},\mathsf{C}] = \langle \mathsf{A},\mathsf{B},\mathsf{C}\rangle$. Then we have PaPB$\models_{\mathbf{MP}_4} \mathrm{D}_{\{\mathsf{B},\mathsf{C}\}}\mathrm{D}_{\mathsf{C}}\mathrm{B}_{\mathsf{A}}p$, for $\{\mathsf{B},\mathsf{C}\} = \{\mathsf{B},\mathsf{C},\mathsf{D}\} \cap (\mathsf{ag@path}[\mathsf{A},\mathsf{C}] \cup \{\mathsf{C}\})$.

PROPOSITION 5. *Given a prov-query $q = D_{AE_n} \cdots D_{AE_1}\phi$ and PaPB, if PaPB$\models_{\mathbf{MP}_4} q$ then for all $1 \le l \le n$, $\mathsf{index}[AE_l, q] \le |AE_l|$.*

Take $q_5 = \mathrm{D}_{\mathsf{B}}\mathrm{D}_{\{\mathsf{B},\mathsf{C}\}}\mathrm{B}_{\mathsf{A}}p$ for instance. It follows that PaPB$\not\models_{\mathbf{MP}_4} q_5$ from Proposition 5.

THEOREM 6. *Given PaPB and a set of distinct agents $\{\mathsf{n}, \mathsf{n}_1, \cdots, \mathsf{n}_l, \mathsf{n}'\}$, the following claims are equivalent:*

1. *$\mathsf{DS}\langle$PaPB$\rangle$ has a path $\langle \mathsf{n}, \mathsf{n}_1, \cdots, \mathsf{n}_l, \mathsf{n}'\rangle$ associated with p.*
2. *PaPB$\models_{\mathbf{MP}_4} D_{\{\mathsf{n}_1,\cdots,\mathsf{n}_l\}} \cdots D_{\{\mathsf{n}_{l-1},\mathsf{n}_l\}} D_{\mathsf{n}_l} T^{\mathsf{n}}_{\mathsf{n}'} p$.*
3. *If $B_{\mathsf{n}'} p \in$ PaPB then PaPB$\models_{\mathbf{MP}_4} D_{\{\mathsf{n}_1,\cdots,\mathsf{n}_l,\mathsf{n}'\}} \cdots D_{\{\mathsf{n}_l,\mathsf{n}'\}} D_{\mathsf{n}'} B_{\mathsf{n}} p$.*

Consider the prov-aware query $q_6 = \mathrm{D}_{\{\mathsf{B},\mathsf{C},\mathsf{D}\}}\mathrm{D}_{\{\mathsf{C},\mathsf{D}\}}\mathrm{D}_{\mathsf{D}}\mathrm{B}_{\mathsf{A}}p$ in Example 5. If PaPB$\models_{\mathbf{MP}_4} q_6$, from Theorem 6 it follows that $\mathsf{DS}\langle$PaPB$\rangle$ has a path $\langle \mathsf{A},\mathsf{B},\mathsf{C},\mathsf{D}\rangle$ associated with p.

To generalize the above-mentioned results, we relax the limitations put on PaPB. Let PO be the set of all policies defined as below.

$$\begin{array}{rcl} \mathsf{policy} & ::= & \mathsf{rule} \mid \mathsf{fact} \\ \mathsf{rule} & ::= & \mathsf{cond} \Rightarrow \mathsf{head} \\ \mathsf{head} & ::= & \mathrm{B}_i p \mid \mathrm{T}^i_j p \\ \mathsf{cond} & ::= & \mathsf{fact} \mid \mathsf{cond} \wedge \mathsf{cond} \\ \mathsf{fact} & ::= & p \mid \mathrm{T}^i_j p \mid \mathrm{B}_i(\mathsf{fact}) \end{array}$$

Given a finite policy base PB such that PB $\subset PO$, we define the set $\mathsf{head}(\text{PB}) = \{h \mid (c_1 \wedge \cdots \wedge c_n \Rightarrow h) \in \text{PB}\}$; let $\mathsf{fact}(\text{PB})$ be the set of facts in PB and $\mathsf{fact}_r(\text{PB}) = \{f \in \mathsf{fact}(\text{PB}) \mid f \in \mathtt{WFF}_{pa}\}$. Let $\mathsf{DS}\langle\text{PB}\rangle$ be $\mathsf{DS}\langle\mathsf{head}(\text{PB}) \cup \mathsf{fact}_r(\text{PB})\rangle$. We refer to a finite policy base PB as *ePaPB* (*extended Provenance aware Policy Bases*) if PB $\subset PO$ and $\mathsf{DS}\langle\text{PB}\rangle$ is acyclic. Given ePaPB, let $\mathsf{activehead}(\text{ePaPB}) = \{h \in \mathsf{head}(\text{ePaPB}) \mid \text{ePaPB} \models_{\mathbf{MP}_4} h\}$.

THEOREM 7. *Given ePaPB, let ePaPB$' = \mathsf{activehead}(ePaPB) \cup \mathsf{fact}_r(ePaPB)$. For a PaPB-query q, ePaPB$' \models_{\mathbf{MP}_4} q$ if and only if ePaPB$\models_{\mathbf{MP}_4} q$.*

As corollaries, the results analogous to Propositions 3 and Theorem 6 can be derived with respect to ePaPB. The intuition is twofold. First of all, the difference between ePaPB and PaPB does not

affect how provenance information is collected. And, on the other hand, PaPB-queries and subclasses thereof concern mostly about provenance.

As a subset of $\mathtt{WFF}_{pa}$, PaPB is still quite expressive. PaPB can be seen as a simplified version of the language used in [4, 3]. Besides, a subset of $\mathtt{WFF}_{pa}$ can be considered as a BT system [20]. As pointed out in [20], "The system BT is useful in modeling a set of cooperative agents in which each agent has unlimited access to other agents' knowledge base." In this work, this reasoning is restricted to a policy base which is constructed from agents' statements. ePaPB, as an extension of PaPB, is more expressive.

6. APPLICATION: SECURITY OF DELEGATION IN LOGIC-BASED POLICY BASES

Wang et al., [24] found that users may circumvent security policies in access control systems using delegations. Here is an illustrative example from [24].

> In a company, the task of issuing checks is modeled by two authorizations *pre* and *app*, which stand for "check preparation" and "approval", respectively. In order to prevent fraudulent transactions, *pre* and *app* must be performed by two *different* members of the role Treasurer (Tr for short). Also, for the sake of resiliency, the company allows a Treasurer to delegate his/her role to a Clerk (Cl for short) in case he/she is not able to work due to sickness or some other reasons. A is a Treasurer and B is a Clerk of the company. They decided to collude to issue checks for themselves.

As noted in [24], A and B are able to issue checks for themselves, through the following actions: (A1) A delegates the role Treasurer to B; (A2) B performs *pre* to prepare a check for A; and (A3) A performs *app* to approve the check prepared by B.

ePaPB is expressive enough to capture this situation as follows.

$$\mathtt{InRole}(\mathsf{A},\mathsf{Tr}) \Rightarrow \mathrm{T}^{\mathsf{L}}_{\mathsf{A}}\mathtt{pre}(\mathit{check}) \quad (19)$$

$$\mathtt{InRole}(\mathsf{A},\mathsf{Tr}) \Rightarrow \mathrm{T}^{\mathsf{L}}_{\mathsf{A}}\mathtt{app}(\mathit{check}) \quad (20)$$

$$\left(\begin{array}{c}\mathtt{InRole}(\mathsf{A},\mathsf{Tr}) \wedge \mathtt{InRole}(\mathsf{B},\mathsf{Cl}) \\ \wedge\, \mathrm{B}_{\mathsf{L}}\mathrm{T}^{\mathsf{A}}_{\mathsf{B}}\mathtt{InRole}(\mathsf{B},\mathsf{Tr})\end{array}\right) \Rightarrow \mathrm{T}^{\mathsf{A}}_{\mathsf{B}}\mathtt{pre}(\mathit{check}) \quad (21)$$

$$\mathtt{InRole}(\mathsf{A},\mathsf{Tr}), \quad \mathtt{InRole}(\mathsf{B},\mathsf{Cl}) \quad (22)$$

$$\mathrm{B}_{\mathsf{L}}\mathrm{T}^{\mathsf{A}}_{\mathsf{B}}\mathtt{InRole}(\mathsf{B},\mathsf{Tr}), \quad \mathrm{B}_{\mathsf{B}}\mathtt{pre}(\mathit{check}), \quad \mathrm{B}_{\mathsf{A}}\mathtt{app}(\mathit{check}) \quad (23)$$

Note that in (21), the formula $\mathrm{B}_{\mathsf{L}}\mathrm{T}^{\mathsf{A}}_{\mathsf{B}}\mathtt{pre}(\mathit{check})$ means that L receives the corresponding credential, verifies the issuers' signature, and thus believes that A does delegate the role Tr to B.

We make queries: $q_1 = \mathrm{D}_{\{\mathsf{A},\mathsf{B}\}}\mathrm{D}_{\mathsf{B}}\mathrm{B}_{\mathsf{L}}\mathtt{pre}(\mathit{check})$, $q_2 = \mathrm{D}_{\mathsf{B}}\mathrm{B}_{\mathsf{L}}\mathtt{pre}(\mathit{check})$, $q_3 = \mathrm{D}_{\{\mathsf{A},\mathsf{B}\}}\mathrm{B}_{\mathsf{L}}\mathtt{pre}(\mathit{check})$, and $q_4 = \mathrm{D}_{\mathsf{A}}\mathrm{B}_{\mathsf{L}}\mathtt{app}(\mathit{check})$. We have ePaPB$\models_{\mathbf{MP}_4} q_1$, which means that the reason why $\mathrm{B}_{\mathsf{L}}\mathtt{pre}(\mathit{check})$ holds is because L delegates the permission $\mathtt{pre}(\mathit{check})$ to A, who further re-delegates to B and B says he wants to prepare the check. One may only concern about whether it is only B who makes the request to prepare the check and causes the authorization reached. Then from ePaPB$\not\models_{\mathbf{MP}_4} q_2$ and ePaPB$\models_{\mathbf{MP}_4} q_3$, we can conclude the authorization $\mathrm{B}_{\mathsf{L}}\mathtt{pre}(\mathit{check})$ is not only due to B but also A. And since ePaPB$\models_{\mathbf{MP}_4} q_4$, A herself is also responsible for $\mathrm{B}_{\mathsf{L}}\mathtt{app}(\mathit{check})$.

One can utilize provenance of the authorizations to enforce the security of delegations. For instance, informed of that *pre*'s provenance is $\{\mathsf{A},\mathsf{B}\}$ and *app*'s provenance is $\{\mathsf{A}\}$, L should detect that the check would be issued simply by one Treasurer A and thus forbid either step. Hence, L is able to defend against provenance-related breaches by checking the authorization provenance.

The security problem with respect to delegation also bothers logic based policy bases. Owing to the unambiguous semantics and formal reasoning ability, logic-based policy bases are common in many distribute systems . These systems range from grid to virtual enterprises. Thus it is important to work out a mechanism to protect these systems from this security problem.

Wang et al., proposed a *Source-based Enforcement Mechanism* (*SEM*) [24]. While effective and efficient in the context of workflows, SEM may fail to live up to expectations of systems where authorization logics are used. First of all, SEM could not work with authorization logics. And, on the other hand, it is not clear how SEM can deal with re-delegation.

We generalize SEM techniques and call this Extended SEM (*ExSEM*). Consider a conference paper review process in Figure 4. Suppose that a conference conf assigns the task of reviewing a paper to two PC member, pc1 and pc2. Accidentally, they both delegates their permissions (i.e., `comment1` and `comment2`, respectively) to a reviewer rv. rv then re-delegates `comment1` and `comment2` to another two reviewers, rv1 and rv2, respectively. This could be regarded as a security threat, since rv has too much control over the result of the paper.

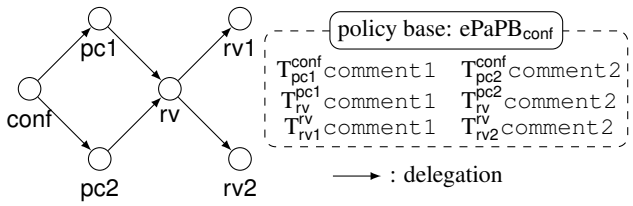

Figure 4: Delegations for reviewing a paper.

When rv1 and rv2 submits `comment1` and `comment2`, respectively (i.e., $\mathrm{B}_{\mathsf{rv1}}\mathtt{comment1}$ and $\mathrm{B}_{\mathsf{rv2}}\mathtt{comment2}$), we have

$$\text{ePaPB}_{\mathsf{conf}} \models_{\mathbf{MP}_4} \mathrm{D}_{\{\mathsf{pc1},\mathsf{rv},\mathsf{rv1}\}}\mathrm{D}_{\{\mathsf{rv},\mathsf{rv1}\}}\mathrm{D}_{\mathsf{rv1}}\mathrm{B}_{\mathsf{conf}}\mathtt{comment1}.$$
$$\text{ePaPB}_{\mathsf{conf}} \models_{\mathbf{MP}_4} \mathrm{D}_{\{\mathsf{pc2},\mathsf{rv},\mathsf{rv2}\}}\mathrm{D}_{\{\mathsf{rv},\mathsf{rv2}\}}\mathrm{D}_{\mathsf{rv2}}\mathrm{B}_{\mathsf{conf}}\mathtt{comment2}.$$

Hence, we could reverse the delegation stories, just as in Figure 4. When conf finds that both rv1 and rv2 are selected by rv, conf may doubt about the justices of the comments and take the provenance of comments into account when making decisions. Generally, various constraints can be checked on provenance and one may evaluate provenance-related properties.

In summary, ExSEM follows the ideas of SEM and generalizes them to the logic-based policy bases. Logic-based policy bases are widely employed in distributed systems. However, SEM itself falls short in this aspect. In ExSEM, DBT can express sources of privileges (as provenance information) and various PaPB-queries can be posed against ePaPB. When a proof of an authorization is presented, ExSEM uses DBT to collect authorization provenance information. This provenance is later checked against various constrains, as in SEM. On the other hand, ExSEM still depends on SEM to verify the relations among the users involved in the provenance and the user who actually requests privileges.

7. RELATED WORKS

DBT builds upon the logic BT [20]. While BT deals with beliefs and trusts, DBT unifies beliefs, trusts and provenance. DBT is

strictly more expressive than BT. Given a rule base RB_{BT} consisting of only BT formulas and a BT query Q_{BT}, we construct a DBT rule base $\mathsf{RB}_{DBT} = \{\mathrm{D}_{\mathcal{AG}}(f) \mid f \in \mathsf{RB}_{BT}\}$, and a DBT query $\mathsf{Q}_{DBT} = \mathrm{D}_{\mathcal{AG}}(\mathsf{Q}_{BT})$, where $\mathcal{AG}$ is the set of all agents involved in RB_{BT}. It can be shown that $BT \vdash \bigwedge \mathsf{RB}_{BT} \Rightarrow \mathsf{Q}_{BT}$ if and only if $DBT_4 \vdash \bigwedge \mathsf{RB}_{DBT} \Rightarrow \mathsf{Q}_{DBT}$. The reason we explore the idea of authorization provenance on BT is that BT supports modular extension because of neighborhood semantics chosen for the trust operator.

Existing access control logics put few emphases on authorization provenance. Roughly, we divide them into modal-logic-based and non-modal-logic-based. The modal-logic-based group works around the treatment of the operators of "says" (i.e. **believes**) and "speak-for" and related properties. Take the logic ABLP [1] for instance. While DBT can express beliefs and delegations, ABLP logic is still more expressive than DBT in terms of authorizations. However, there is no operator designed to capture authorization provenance, which is the focus of DBT. Since they are interpreted in the same framework as DBT, it seems feasible to extend ABLP logic to express provenance or to build a new logic on them. One may define operators for provenance and impose some reasonable relations among modalities.

The non-modal-logic-based line includes FAF [14], Binder [9], SecPAL [5], DL [18], and RT [19], to name a few. In general, these languages achieve a balance between the expressiveness and computational tractability. Nevertheless, none of these policy languages focuses on what the operator D_i is designed to capture.

Still, some notions similar to provenance is proposed in literature. SD3 [16] produces a *proof tree* along with the answer to each query. The proof tree is used to checked the correctness of the proof. One can not ask if a conclusion with a specific provenance is true in SD3 policy bases. RT_0 [19] forms delegation chains for a policy base. Since its focus is on how to store and retrieve credentials in a distribute way, it can hardly answer queries about provenance. For example, suppose that both "B causes A to believe a fact **f**" and "C causes A to believe **f**" are true. RT_0 only handles queries of "A believes **f**", without telling the difference of B and C. SecPAL supports several bounded delegations such as depth-bounded delegations and width bounded delegation [5]. These bounded delegation is effective for prohibiting delegation with certain provenance from taking effects. However, SecPAL could not specify formulas that include provenance, thus losing the advantages of reasoning about provenance.

Though mechanisms CSE [21] and SEM [24] proposed by Mao et al., and Wang et al., respectively, defend against attacks resulting from neglect of provenance, they fall short when working with logic-based policy bases. Nevertheless, the approaches in this paper rely on CSE and SEM to tracing the contamination source or verify relationships among agents appearing in provenance.

Vaughan et al. motivated and presented a framework to log proofs of authorizations for auditing [23]. While detailed analysis of these logs may help detect flaws in complex authorization policies, an implicit assumption is made that the whole proof of an authorization be available to logging. However, under certain circumstances, it appears demanding, for the whole proof may be at the third party, difficult to obtain, or refused to be accessed for privacy reasons. An entry of the conclusion encapsulating both authorization and its provenance is easier to ship and store. The provenance-enabled conclusions also provide useful information for analyzing improper authorizations.

8. CONCLUSION

We have presented the motivations, design, and applications of a provenance-enabled authorization logic, DBT. DBT extends the BT logic mostly by introducing the operator D_i, and integrates beliefs, trusts, and provenance in a unified framework. DBT achieves some benefits: (1) defense against a type of delegation-exploiting attacks at the access control level, (2) understanding and analysis of authorizations and the status of policy bases, and (3) potentially efficient auditing guided by provenance information.

There are several avenues for future work. Since the operator of "speak-for" plays an important role distributed authorizations, we are in the process of extending DBT to capture the provenance when working with "speak-for". In addition, the notion of role is useful for authorization, we also plan to support provenance in presence of roles in authorization logics.

Acknowledgment

This work is supported by National Natural Science Foundation of China under Grant 60873225, 60773191, 70771043, National High Technology Research and Development Program of China under Grant 2007AA01Z403, Natural Science Foundation of Hubei Province under Grant 2009CDB298, Open Foundation of State Key Laboratory of Software Engineering under Grant SKLSE20080718, and Innovation Fund of Huazhong University of Science and Technology under Grant Q2009021. This project is supported in part by an Australian Research Council (ARC) Discovery Projects Grant (DP0988396). We thank the anonymous reviewers for their helpful comments.

9. REFERENCES

[1] M. Abadi, M. Burrows, B. W. Lampson, and G. D. Plotkin. A calculus for access control in distributed systems. *ACM Trans. Program. Lang. Syst.*, 15:706–734, 1993.

[2] A. W. Appel and E. W. Felten. Proof-carrying authentication. In *ACM Conference on Computer and Communications Security*, pages 52–62, 1999.

[3] L. Bauer, S. Garriss, and M. K. Reiter. Distributed proving in access-control systems. In *IEEE Symposium on Security and Privacy*, pages 81–95, 2005.

[4] L. Bauer, S. Garriss, and M. K. Reiter. Efficient proving for practical distributed access-control systems. In *ESORICS*, pages 19–37, 2007.

[5] M. Y. Becker, C. Fournet, and A. D. Gordon. Design and semantics of a decentralized authorization language. In *CSF*, pages 3–15, 2007.

[6] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[7] P. Bonatti and P. Samarati. Logics for authorizations and security. In J. Chomicki, R. van der Meyden, and G. Saake, editors, *Logics for Emerging Applications of Databases*. Springer-Verlag, 2003.

[8] S. B. Davidson and J. Freire. Provenance and scientific workflows: challenges and opportunities. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1345–1350, 2008.

[9] J. DeTreville. Binder, a logic-based security language. In *IEEE Symposium on Security and Privacy*, pages 105–113, 2002.

[10] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[11] R. Fagin, J. Y. Halpern, and M. Y. Vardi. What can machines know? on the properties of knowledge in distributed systems. In *Journal of the ACM*, volume 39, pages 328–376, 1992.

[12] D. Garg and M. Abadi. A modal deconstruction of access control logics. In *Foundations of Software Science and Computation Structures*, pages 216–230, 2008.

[13] J. Y. Halpern and V. Weissman. Using first-order logic to reason about policies. In *IEEE Computer Security Foundations Symposium*, pages 187–201, 2003.

[14] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian. Flexible support for multiple access control policies. *ACM Trans. Database Syst.*, 26(2):214–260, 2001.

[15] X. Jiang, A. Walters, D. Xu, E. H. Spafford, F. Buchholz, and Y.-M. Wang. Provenance-aware tracing ofworm break-in and contaminations: A process coloring approach. In *ICDCS*, page 38, 2006.

[16] T. Jim. SD3: A trust management system with certified evaluation. In *IEEE Symposium on Security and Privacy*, pages 106–115, 2001.

[17] C. Lesniewski-Laas, B. Ford, J. Strauss, R. Morris, and M. F. Kaashoek. Alpaca: extensible authorization for distributed services. In *14th ACM conference on Computer and communications security*, pages 432–444, 2007.

[18] N. Li, B. N. Grosof, and J. Feigenbaum. Delegation logic: A logic-based approach to distributed authorization. *ACM Trans. Inf. Syst. Secur.*, 6(1):128–171, 2003.

[19] N. Li, W. H. Winsborough, and J. C. Mitchell. Distributed credential chain discovery in trust management. *Journal of Computer Security*, 11(1):35–86, Feb. 2003.

[20] C.-J. Liau. Belief, information acquisition, and trust in multi-agent systems - a modal logic formulation. *Artificial Intelligence*, 149:31–60, 2003.

[21] Z. Mao, N. Li, H. Chen, and X. Jiang. Trojan horse resistant discretionary access control. In *SACMAT*, 2009.

[22] W. C. Tan. Provenance in databases: Past, current, and future. *IEEE Data Eng. Bull.*, 30(4):3–12, 2007.

[23] J. A. Vaughan, L. Jia, K. Mazurak, and S. Zdancewic. Evidence-based audit. In *22nd IEEE Computer Security Foundations Symposium*, pages 177–191, 2008.

[24] Q. Wang, N. Li, and H. Chen. On the security of delegation in access control systems. In *ESORICS*, pages 317–332, 2008.

APPENDIX

A. PROOF SKETCH OF THEOREM 1

We prove for the case where $t = 4$, i.e., with respect to $\mathbf{MP}_4$ and DBT_4. It is straightforward to check that the axioms in DBT_4 (Figure 1) are valid, and that the inference rules also preserve the validity. We proceeds to the proof of completeness. To prove the completeness, we use the canonical model method [6], while some techniques used in [10, 11, 20] are also borrowed.

A wff φ is DBT_4-*consistent* if its negation $\neg\varphi$ can not be proved in DBT_4. A finite set $\{\varphi_1, \cdots, \varphi_n\}$ of wff is DBT_4-consistent if $\varphi_1 \wedge \cdots \wedge \varphi_n$ is DBT_4-consistent, and an infinite set of formulas is DBT_4-consistent if all of its finite subsets are DBT_4-consistent. A set V of wff is a maximal DBT_4-consistent set if it is DBT_4-consistent and for all wff φ not in V the set $V \cup \{\varphi\}$ is not DBT_4-consistent. On the other hand, φ is satisfiable iff there is a model $\mathcal{M}$ in $\mathbf{MP}_4$ and a world w such that $\langle \mathcal{M}, w\rangle \models \varphi$.

A canonical structure $\mathcal{M}^*$ is a tuple $\langle W, \pi, \mathcal{B}_i, \mathcal{D}_i, \mathcal{T}_j^i, \mathcal{D}_{AE}\rangle$ $(i, j \in \mathcal{AG}, i \neq j)$ where (2) $W = \{w_V \mid V \text{ is a maximal } DBT_4\text{-consistent set.}\}$ (2) $\pi : W \mapsto 2^{\texttt{Prop}}$ is defined as $\pi(w_V) = \{p \in \texttt{Prop} \mid p \in V\}$. (3) $\mathcal{B}_i(w_V, w_U)$ iff $V/\mathrm{B}_i \subseteq U$, where $V/\mathrm{B}_i = \{\varphi \mid \mathrm{B}_i\varphi \in V\}$. (4) $\mathcal{D}_i(w_V, w_U)$ iff $V/\mathrm{D}_i \subseteq U$, where $V/\mathrm{D}_i = \{\varphi \mid \mathrm{D}_i\varphi \in V\}$. (5) $\mathcal{D}_{AE}(w_V, w_U)$ iff $V/\mathrm{D}_{AE} \subseteq U$, where $V/\mathrm{D}_{AE} = \{\varphi \mid \mathrm{D}_{AE}\varphi \in V\}$. And (6) $\mathcal{T}_j^i(w_V) = \{(\!|\varphi|\!) \mid \mathrm{T}_j^i\varphi \in V\}$, where $(\!|\varphi|\!) = \{w_U \in W \mid \varphi \in U\}$.

Let $\mathcal{M}$ be an $\mathbf{MP}_4$ model. Define $\langle \mathcal{M}^*, w\rangle \models \varphi$ just as $\langle \mathcal{M}, w\rangle \models \varphi$ in definition 1, except that $\langle \mathcal{M}^*, w\rangle \models \mathrm{D}_{AE}\varphi$ iff $\langle \mathcal{M}^*, w'\rangle \models \varphi$ for all w' such that $(w, w') \in \mathcal{D}_{AE}$. It can be shown that every DBT_4-consistent wff is satisfiable in $\mathcal{M}^*$. We first prove this fact, and then transform $\mathcal{M}^*$ to an $\mathbf{MP}_4$ model $\mathcal{M}_\sharp$ such that $\langle \mathcal{M}^*, w\rangle \models \varphi$ iff there exists a possible world $w_\sharp$ such that $\langle \mathcal{M}_\sharp, w_\sharp\rangle \models \varphi$.

We now prove that, for any wff φ and any maximal DBT_4-consistent set V, $\langle \mathcal{M}^*, s_V\rangle \models \varphi$ iff $\varphi \in V$ by induction on the structure of φ. We only show the case where φ is $\mathrm{T}_j^i\psi$ for example. First assume $\langle \mathcal{M}^*, s_V\rangle \models \mathrm{T}_j^i\psi$. Then by the definition of $\models$, we have $\lceil\psi\rceil \in \mathcal{T}_j^i(s_V)$. That is, the set $\{s_U \mid \langle \mathcal{M}^*, s_U\rangle \models \psi\} \in \mathcal{T}_j^i(s_V)$. It follows that, by induction hypothesis, $\{s_U \mid \psi \in U\} \in \mathcal{T}_j^i(s_V)$; that is, $(\!|\psi|\!) \in \mathcal{T}_j^i(s_V)$. According to the definition of $\mathcal{T}_j^i$ in $\mathcal{M}^*$, there exists ϕ such that $\mathrm{T}_j^i\phi \in V$ and $(\!|\psi|\!) = (\!|\phi|\!)$. Then we have $DBT_4 \vdash \psi \Leftrightarrow \phi$. For suppose otherwise. Either $\neg\psi \wedge \phi$ or $\psi \wedge \neg\phi$ is DBT_4-consistent; therefore there exists a maximal DBT_4-consistent set containing either ψ or ϕ, but not both. It follows that $(\!|\psi|\!) \neq (\!|\phi|\!)$, leading to a contradiction. Since $DBT_4 \vdash \psi \Leftrightarrow \phi$, $DBT_4 \vdash \mathrm{T}_j^i\psi \Leftrightarrow \mathrm{T}_j^i\phi$ holds. And from $\mathrm{T}_j^i\phi \in V$, we have $\mathrm{T}_j^i\psi \in V$, i.e., $\varphi \in V$. For the other direction, assume that $\mathrm{T}_j^i\psi \in V$. Then $(\!|\psi|\!) \in \mathcal{T}_j^i(s_V)$, from which, by the induction hypothesis, it follows that $\lceil\psi\rceil \in \mathcal{T}_j^i(s_V)$. Hence, $\langle \mathcal{M}^*, s_V\rangle \models \mathrm{T}_j^i\psi$.

Next, we can prove that $\mathcal{B}_i$ is a serial, transitive and Euclidean binary relation. We now verify that $\mathcal{M}^*$ satisfies the constraints from C_{SBT} to $C_{\text{AE-Red}}$. We only show the case C_{SRB}: $\mathcal{B}_i(w) = \mathcal{D}_i \circ \mathcal{B}_i(w)$ for example. First, we show that $\mathcal{D}_i \circ \mathcal{B}_i(s_V) \subseteq \mathcal{B}_i(s_V)$. Suppose that $s_U \in \mathcal{D}_i \circ \mathcal{B}_i(s_V)$; then there is a state s_Y such that $(s_V, s_Y) \in \mathcal{D}_i$ and $(s_Y, s_U) \in \mathcal{B}_i$. For any $\mathrm{B}_i\varphi \in V$, by axiom SRB, we have $\mathrm{D}_i\mathrm{B}_i\varphi \in V$. Since $V/\mathrm{D}_i \subseteq Y$, $\mathrm{B}_i\varphi \in Y$ and, in turn, $\varphi \in U$. Hence, $V/\mathrm{B}_i \subseteq U$; $s_U \in \mathcal{B}_i(s_V)$ holds.

We now show $\mathcal{B}_i(s_V) \subseteq \mathcal{D}_i \circ \mathcal{B}_i(s_V)$. Assume that $s_U \in \mathcal{B}_i(s_V)$, i.e., $V/\mathrm{B}_i \subseteq U$. Define $V/\mathrm{D}_i\mathrm{B}_i$ as the set $\{\varphi \mid \mathrm{D}_i\mathrm{B}_i\varphi \in V\}$. For any $\mathrm{D}_i\mathrm{B}_i\varphi \in V$, from the axiom SRB and V being a maximal DBT_4-consistent set, $\mathrm{B}_i\varphi \in V$. Then from the assumption, it follows that $\varphi \in U$, i.e., $V/\mathrm{D}_i\mathrm{B}_i \subseteq U$. We have to show that there is a maximal DBT_4-consistent set Y such that $\mathcal{D}_i(s_V, s_Y)$ and $\mathcal{B}_i(s_Y, s_U)$. If the set Δ, $\{\phi \mid \mathrm{D}_i\phi \in V\} \cup \{\neg\mathrm{B}_i\neg\psi \mid \psi \in U\}$, is consistent, we could obtain such a Y by extending Δ. Suppose that Y is a maximal DBT_4-consistent set containing Δ; then since for any $\mathrm{D}_i\phi \in V$ $\phi \in Y$, we have $V/\mathrm{D}_i \subseteq Y$. We can also show that $Y/\mathrm{B}_i \subseteq U$. Suppose for the sake of contradiction that there exists some $\mathrm{B}_i\varphi \in Y$ but $\varphi \notin U$. Then, $\neg\varphi \in U$, from the construction of Δ, it follows that $\neg\mathrm{B}_i\varphi \in Y$, a contradiction with the assumption $\mathrm{B}_i\varphi \in Y$. Hence, $\mathcal{D}_i(s_V, s_Y)$ and $\mathcal{B}_i(s_Y, s_U)$ hold.

So all we need to do is to verify that Δ is consistent. Suppose for the sake of contradiction that Δ is not consistent. Then there exist $\phi_1, \cdots, \phi_m, \psi_1, \cdots, \psi_n$ such that $DBT_4 \vdash (\phi_1 \wedge \cdots \wedge \phi_m \wedge \neg\mathrm{B}_i\neg\psi_1 \wedge \cdots \wedge \mathrm{B}_i\neg\psi_n) \Rightarrow \bot$. Let $\hat{\phi}$ be $\phi_1 \wedge \cdots \wedge \phi_m$ and $\hat{\psi}$ be $\psi_1 \wedge \cdots \wedge \psi_n$. Note that $\hat{\psi} \in U$. Then we have $DBT_4 \vdash \neg\mathrm{B}_i\neg\hat{\psi} \Rightarrow \neg\mathrm{B}_i\neg\psi_1 \wedge \cdots \wedge \mathrm{B}_i\neg\psi_n$, from which it follows that $DBT_4 \vdash \hat{\phi} \wedge \neg\mathrm{B}_i\neg\hat{\psi} \Rightarrow \bot$. So $DBT_4 \vdash \hat{\phi} \Rightarrow \mathrm{B}_i\neg\hat{\psi}$; by axiom D1 and rule R2, $DBT_4 \vdash \mathrm{D}_i\hat{\phi} \Rightarrow \mathrm{D}_i\mathrm{B}_i\neg\hat{\psi}$. Since $\mathrm{D}_i\phi_1, \cdots, \mathrm{D}_i\phi_m \in V$, we have $\mathrm{D}_i\hat{\phi} \in V$, from which $\mathrm{D}_i\mathrm{B}_i\neg\hat{\psi} \in V$ can be derived. And we have shown that $\{\varphi \mid \mathrm{D}_i\mathrm{B}_i\varphi \in V\} \subseteq U$. Hence, $\neg\hat{\psi} \in U$. However, $\hat{\psi} \in U$ also holds, a contradiction with U being a maximal DBT_4-consistent set.

We now transform $\mathcal{M}^*$ into an $\mathbf{MP}_4$ model through an intermediate Kripke structure. Before doing this, we introduce some useful notions [11].

Given a Kripke structure $\mathcal{M}^o = \langle W^o, \pi^o, \mathcal{B}_i^o, \mathcal{D}_i^o, \mathcal{T}_j^{io}\rangle$ $(i, j \in \{1, \cdots, m\}$ and $i \neq j)$ and $s, t \in W^o$, we say a sequence $\langle v_1, i_1, v_2, i_2, \cdots, i_{k-1}, v_k\rangle$ where $k \geq 1$ is a *D-path from s to t* if (1) $v_1 = s$, (2) $v_k = t$, (3) $v_1, \cdots, v_k$ are states, (4) $i_1, \cdots, i_{k-1}$ are agents, and (5) $(v_l, v_{l+1}) \in \mathcal{D}_{i_l}^o$, for $1 \leq l < k$. A D-path $\langle v_1, i_1, v_2, i_2, \cdots, i_{k-1}, v_k\rangle$ is *reduced* if $i_l \neq i_{l+1}$ for $1 \leq l < k$. We say that a structure $\mathcal{M}^o$ is *D-tree-like* if whenever s and t are states of $\mathcal{M}^o$, then there is at most one D-path from s to t in $\mathcal{M}^*$.

We now construct a model $\overline{\mathcal{M}} = \{\overline{W}, \overline{\pi}, \overline{\mathcal{B}_i}, \overline{\mathcal{D}_i}, \overline{\mathcal{T}_j^i}, \overline{\mathcal{D}_{AE}}\}$ from $\mathcal{M}^* = \langle W, \pi, \mathcal{B}_i, \mathcal{D}_i, \mathcal{T}_j^i, \mathcal{D}_{AE}\rangle$. Let $L_1 = W$. Assuming L_k has been defined, for each $s \in W$, each $v \in L_k$, and each l (l may be an agent i or AE), we define a new state $z_{s,v,l}$, and refer to $z_{s,v,l}$ as an *l-child* of v. L_{k+1} consists of all these states $z_{s,v,l}$. Then let $\overline{W} = \bigcup\{L_k \mid k \geq 1\}$. Define $g : \overline{W} \mapsto W$ by letting $g(s) = s$ if $s \in L_1$, and $g(z_{s,v,l}) = s$ for $z_{s,v,l} \in L_k$ where $k \geq 2$. Define $\overline{\pi}$, $\overline{\mathcal{B}_i}$, $\overline{\mathcal{T}_j^i}$, $\overline{\mathcal{D}_i}$, and $\overline{\mathcal{D}_{AE}}$, respectively, as follows: (1) $\overline{\pi}(s) = \pi(g(s))$ (2) $(s, t) \in \overline{\mathcal{B}_i}$ iff $(g(s), g(t)) \in \mathcal{B}_i$; (3) $(s, W_t) \in \overline{\mathcal{T}_j^i}$ iff $(g(s), W_t^g) \in \mathcal{T}_j^i$ where $W_t^g = \{g(t) \mid t \in W_t\}$; (4) define $\overline{\mathcal{D}_i}$ by letting $(s, t) \in \overline{\mathcal{D}_i}$ iff t is an *i-child* of s and $(g(s), g(t)) \in \mathcal{D}_i$; (5) define $\overline{\mathcal{D}_{AE}}$ by letting $(s, t) \in \overline{\mathcal{D}_{AE}}$ iff t is an *AE-child* of s and $(g(s), g(t)) \in \mathcal{D}_{AE}$. As shown in [11], $\overline{\mathcal{M}}$ is D-tree-like, and $\langle \overline{\mathcal{M}}, s\rangle \models \varphi$ iff $\langle \mathcal{M}^*, g(s)\rangle \models \varphi$ when φ does not take the form of $\varphi = \mathrm{B}_i\psi$ or $\varphi = \mathrm{T}_j^i\psi$. We can also show that it is also the case when $\varphi = \mathrm{B}_i\psi$ or $\varphi = \mathrm{T}_j^i\psi$. Put together, $\overline{\mathcal{M}}$ is D-tree-like, and $\langle \overline{\mathcal{M}}, s\rangle \models \varphi$ iff $\langle \mathcal{M}^*, g(s)\rangle \models \varphi$.

Based on the structure $\overline{\mathcal{M}} = \{\overline{W}, \overline{\pi}, \overline{\mathcal{B}_i}, \overline{\mathcal{D}_i}, \overline{\mathcal{T}_j^i}, \overline{\mathcal{D}_{AE}}\}$, we can obtain an $\mathbf{MP}_4$ model $\mathcal{M}_\sharp = \langle W_\sharp, \pi_\sharp, \mathcal{B}_{i_\sharp}, \mathcal{D}_{i_\sharp}, \mathcal{T}_{j_\sharp}^i\rangle$ by letting (1) $W_\sharp$ be $\overline{W}$, (2) $\pi_\sharp$ be $\overline{\pi}$, (3) $\mathcal{B}_{i_\sharp}$ be $\overline{\mathcal{B}_i}$, (4) $\mathcal{D}_{i_\sharp}$ be $\overline{\mathcal{D}_i} \cup \overline{\mathcal{D}_{AE}}$, and (5) $\mathcal{T}_{j_\sharp}^i$ be $\overline{\mathcal{T}_j^i}$. As shown in [11], it can be proved that $\langle \overline{\mathcal{M}}, s\rangle \models \varphi$ iff $\langle \mathcal{M}_\sharp, s\rangle \models \varphi$, when φ does not take the form of $\mathrm{B}_i\psi$ or $\mathrm{T}_j^i\psi$. However, for the cases where φ is $\mathrm{B}_i\psi$ or $\mathrm{T}_j^i\psi$, since $\mathcal{B}_{i_\sharp} = \overline{\mathcal{B}_i}$ and $\mathcal{T}_{j_\sharp}^i = \overline{\mathcal{T}_j^i}$, this conclusion holds as well.

From the construction of $\mathcal{M}_\sharp$, it is not hard to see that each $\mathcal{B}_{i_\sharp}$ is a serial, tran-

sitive and Euclidean binary relation. Since $\langle \mathcal{M}^*, g(s)\rangle \models \varphi$ iff $\langle \overline{\mathcal{M}}, s\rangle \models \varphi$ iff $\langle \mathcal{M}_\sharp, s\rangle \models \varphi$, and $\mathcal{M}^*$ satisfies all the constraints from C_{SRT} to $C_{\text{AE-Red}}$, $\mathcal{M}_\sharp$ also satisfies these constraints. For example, suppose, for the sake of contradiction, that $\mathcal{M}_\sharp$ does not comply with C_{Dlgt}. Then there is a state s such that $\langle \mathcal{M}_\sharp, s\rangle \models \mathrm{T}^i_j\varphi$, $\langle \mathcal{M}_\sharp, s\rangle \models \mathrm{B}_j\varphi$, but $\langle \mathcal{M}_\sharp, s\rangle \not\models \mathrm{D}_j\mathrm{B}_i\varphi$. It follows that $\langle \mathcal{M}^*, g(s)\rangle \models \mathrm{T}^i_j\varphi$, $\langle \mathcal{M}^*, g(s)\rangle \models \mathrm{B}_j\varphi$, but $\langle \mathcal{M}^*, g(s)\rangle \not\models \mathrm{D}_j\mathrm{B}_i\varphi$. However, since $\mathcal{M}^*$ obeys C_{Dlgt}, $\langle \mathcal{M}^*, g(s)\rangle \models \mathrm{T}^i_j\varphi \wedge \mathrm{B}_j\varphi \Rightarrow \mathrm{D}_j\mathrm{B}_i\varphi$, a contraction. For $C_{\text{AE-Red}}$ to hold, it requires that $\overline{\mathcal{D}_i} \cap \overline{\mathcal{D}_j} = \emptyset$ for any $i \neq j$. This is because in this case $\mathcal{D}_{AE_\sharp} =$ (by definition) $\bigcap_{i\in AE} \mathcal{D}_{i\sharp} = \overline{\mathcal{D}_{AE}}$. Suppose, for the sake of contradiction, that $(s,t) \in \overline{\mathcal{D}_i} \cap \overline{\mathcal{D}_j}$. Then according to the definitions of $\overline{\mathcal{D}_i}$ and $\overline{\mathcal{D}_j}$, t is both the *i-child* and *j-child* of s. Then $i = j$, a contradiction. Hence, $\mathcal{M}_\sharp$ is an $\mathbf{MP}_4$ model. This completes the proof.

B. PROOF SKETCHES IN SECTION 5

From Corollary 2, in this appendix, we write $DBT_4 \vdash (\bigwedge \text{PaPB} \Rightarrow \phi)$ and $\text{PaPB}\models_{\mathbf{MP}_4}\phi$ interchangeably. We only prove the case where ϕ is of the form $\mathrm{T}^i_j p$; the proof for the case where ϕ is of the form $\mathrm{B}_i p$ is similar. When proving by induction, we omit the base case because it is quite obvious.

Proof Sketch of Proposition 3.

The first claim is obvious because all delegatees are recorded in provenance. We prove the second claim by induction on the length h of the proof of $\text{PaPB}\models_{\mathbf{MP}_4} q$.

Base Case: $h = 2$. Since the proof is of two lines, the instances of axioms used are simply SRT and i-centric-Dlgt. In the case of SRT, assuming that $\mathrm{D}_i\mathrm{T}^i_j p$ is derived, $i \in \mathsf{ag@path}[i,j]$. And for the case of i-centric-Dlgt, assuming that $\mathrm{D}_k\mathrm{T}^i_j p$, $k \in \mathsf{ag@path}[i,j]$, because of $h = 2$.

Course-of-values inductive step: Assume that the proof has length $h+1$ and the statement is true for all numbers less than $h+1$. We enumerate all the cases how $\text{PaPB}\models_{\mathbf{MP}_4} q$ can be concluded.

Case SRT. Since $AE = \{i\} \subseteq \mathsf{ag@path}[i,j]$ and from the induction hypothesis, the statement holds.

Case D3. Suppose that an instance of D3 is applied to $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE_n}\cdots\mathrm{D}_{\widetilde{AE_l}}\cdots\mathrm{D}_{AE_1}\mathrm{T}^i_j p$ at step $h+1$. From the induction hypothesis, it holds that $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE'_n}\cdots\mathrm{D}_{\widetilde{AE'_l}}\cdots\mathrm{D}_{AE'_1}\mathrm{T}^i_j p$. By the instance of D3: $\mathrm{D}_{\widetilde{AE'_l}}\varphi \Rightarrow \mathrm{D}_{\widetilde{AE'_l}\cup(AE_l\cap(\mathsf{ag@path}[i,j]\cup\{j\}))}\varphi$, one obtains $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE'_n}\cdots\mathrm{D}_{AE'_l}\cdots\mathrm{D}_{AE'_1}\mathrm{T}^i_j p$.

Case AE-Red. Suppose that an instance of AE-Red is applied to $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE_n}\cdots\mathrm{D}_{AE_l}\mathrm{D}_{AE_l}\cdots\mathrm{D}_{AE_1}\mathrm{T}^i_j p$. From the induction hypothesis, $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE'_n}\cdots\mathrm{D}_{AE'_l}\mathrm{D}_{AE'_l}\cdots\mathrm{D}_{AE'_1}\mathrm{T}^i_j p$. Then by applying the same instance, we have $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE'_n}\cdots\mathrm{D}_{AE'_l}\cdots\mathrm{D}_{AE'_1}\mathrm{T}^i_j p$.

Case i-centric-Dlgt. Suppose the axiom is applied to $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE_n}\cdots\mathrm{D}_{AE_2}(\mathrm{T}^i_{\mathsf{n}_l} p \wedge \mathrm{T}^{\mathsf{n}_l}_j p)$. Note that in this case $AE_1 = \{\mathsf{n}_l\}$. Then it holds at the step less than h that $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE_n}\cdots\mathrm{D}_{AE_2}\mathrm{T}^i_{\mathsf{n}_l} p$ and $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE_n}\cdots\mathrm{D}_{AE_2}\mathrm{T}^{\mathsf{n}_l}_j p$. From the inductive hypothesis, we have $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{\overline{AE_n}}\cdots\mathrm{D}_{\overline{AE_2}}\mathrm{T}^i_{\mathsf{n}_l} p$, where $\overline{AE_t} = AE_t \cap (\mathsf{ag@path}[i,\mathsf{n}_l]\cup\{\mathsf{n}_l\})$ for $2 \le t \le n$ and $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{\underline{AE_n}}\cdots\mathrm{D}_{\underline{AE_2}}\mathrm{T}^{\mathsf{n}_l}_j p$, where $\underline{AE_t} = AE_t \cap (\mathsf{ag@path}[\mathsf{n}_l,j]\cup\{j\})$ for $2 \le t \le n$. It follows that $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{\underline{AE_n}\cup\overline{AE_n}}\cdots\mathrm{D}_{\underline{AE_2}\cup\overline{AE_2}}\mathrm{T}^i_{\mathsf{n}_l} p \wedge \mathrm{T}^{\mathsf{n}_l}_j p$ by applying the axiom D3. By the axiom i-centric-Dlgt, we have $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{\underline{AE_n}\cup\overline{AE_n}}\cdots\mathrm{D}_{\underline{AE_2}\cup\overline{AE_2}}\mathrm{D}_{\{\mathsf{n}_l\}}\mathrm{T}^i_j p$. For $2 \le t \le n$, from the construction of $\overline{AE_t}$ and $\underline{AE_t}$, it holds that $\overline{AE_t} \cup \underline{AE_t} = AE_t \cap (\mathsf{ag@path}[i,j]\cup\{j\})$. In addition, $\mathsf{n}_l \in \mathsf{ag@path}[i,j]$. Hence, the statement is true when the length of the prove is $h+1$.

Proof Sketch of Proposition 5.

We prove by induction on the length h of the proof of $\text{PaPB}\models_{\mathbf{MP}_4} q$.

Base Case: $h = 2$. Since PaPB consists only of formulas $\mathrm{B}_i p_1$ and $\mathrm{T}^j_k p_2$, the derivation of $\text{PaPB}\models_{\mathbf{MP}_4} q$ are reached by an application of the axiom i-centric-Dlgt. However, the application of the axiom i-centric-Dlgt leads to $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_k\mathrm{T}^i_j p$. We have $\mathsf{index}[\{k\},q] \le |\{k\}| = 1$. The reason why SRT is not applied is that q is a prov-query (i.e., $\mathsf{issuer}[\phi] = i \notin \mathsf{prov}[q]$).

Course-of-values inductive step: Assume that the proof has length $h+1$ and the statement is true for all numbers less than $h+1$.

Case D3. Suppose that an instance of D3: $\mathrm{D}_{AE'_l}\cdots\mathrm{D}_{AE_1}\phi \Rightarrow \mathrm{D}_{AE_l}\cdots\mathrm{D}_{AE_1}\phi$ where $AE'_l \subseteq AE_l$ is applied to $\text{PaPB}\models_{\mathbf{MP}_4}\varphi$, where $\varphi = \mathrm{D}_{AE_n}\cdots\mathrm{D}_{AE'_l}\cdots\mathrm{D}_{AE_1}\phi$. From the induction hypothesis, $\mathsf{index}[AE_t,\varphi] \le |AE_t|$ for $1 \le t \le n$ but $t \neq l$, and $\mathsf{index}[AE'_l,\varphi] \le |AE'_l|$. However, since $AE'_l \subseteq AE_l$, $\mathsf{index}[AE_l,q] = \mathsf{index}[AE'_l,\varphi] \le |AE'_l| \le |AE_l|$. In addition, for $1 \le t \le n$ but $t \neq l$, $\mathsf{index}[AE_t,q] = \mathsf{index}[AE_t,\varphi] \le |AE_t|$.

Case AE-Red. Suppose that an instance of AE-Red: $\mathrm{D}_{AE_l}\mathrm{D}_{AE_l}\cdots\mathrm{D}_{AE_1}\phi \Rightarrow \mathrm{D}_{AE_l}\cdots\mathrm{D}_{AE_1}\phi$ is applied to $\text{PaPB}\models_{\mathbf{MP}_4}\varphi$, where $\varphi = \mathrm{D}_{AE_n}\cdots\mathrm{D}_{AE_l}\mathrm{D}_{AE_l}\cdots\mathrm{D}_{AE_1}\phi$. From the induction hypothesis, $\mathsf{index}[AE_t] \le |AE_t|$ for $1 \le t \le n$. However, for, $1 \le t \le n$, either $\mathsf{index}[AE_t,q] \le \mathsf{index}[AE_t,\varphi]$ or $\mathsf{index}[AE_t,q] = \mathsf{index}[AE_t,\varphi]$; in both cases, $\mathsf{index}[AE_t,q] \le |AE_t|$.

Case i-centric-Dlgt. For an instance of i-centric-Dlgt to be applied, there is some step at the length less then $h+1$ in the proof such that $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE_n}\cdots\mathrm{D}_{AE_2}(\mathrm{T}^i_j p \wedge \mathrm{T}^j_k p)$. So, $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE_n}\cdots\mathrm{D}_{AE_2}\mathrm{T}^i_j p$ and $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE_n}\cdots\mathrm{D}_{AE_2}\mathrm{T}^j_k p$ holds at some step of length less than $h+1$. Denote $\varphi_1 = \mathrm{D}_{AE_n}\cdots\mathrm{D}_{AE_2}\mathrm{T}^i_j p$ and $\varphi_2 = \mathrm{D}_{AE_n}\cdots\mathrm{D}_{AE_2}\mathrm{T}^j_k p$. From the induction hypothesis, for $2 \le t \le n$, $\mathsf{index}[AE_t,\varphi_1] = \mathsf{index}[AE_t,\varphi_2] \le |AE_t|$. If $\mathsf{index}[AE_t,\varphi_1] = \mathsf{index}[AE_t,\varphi_2] < |AE_t|$ for $2 \le t \le n$, then the statement is true when the proof is of length $h+1$. This is because $\mathsf{index}[AE_t,q] = \mathsf{index}[AE_t,\varphi_1] + 1 \le |AE_t|$ and $\mathsf{index}[AE_1,q] = \mathsf{index}[\{\mathsf{n}_l\},q] = |\{\mathsf{n}_l\}| = 1$.

Now assume that for some t such that $2 \le t \le n$, we have $\mathsf{index}[AE_t,\varphi_1] = \mathsf{index}[AE_t,\varphi_2] = |AE_t|$. Let $\overline{\varphi_1}$ be $\mathrm{D}_{AE_n}\cdots\mathrm{D}_{\overline{AE_t}}\cdots\mathrm{D}_{AE_2}\mathrm{T}^i_j p$, where $\overline{AE_t} = AE_t \cap (\mathsf{ag@path}[i,j]\cup\{j\})$, and $\underline{\varphi_2}$ be $\mathrm{D}_{AE_n}\cdots\mathrm{D}_{\underline{AE_t}}\cdots\mathrm{D}_{AE_2}\mathrm{T}^i_j p$, where $\underline{AE_t} = AE_t \cap (\mathsf{ag@path}[j,k]\cup\{k\})$ From Proposition 3, $\text{PaPB}\models_{\mathbf{MP}_4}\overline{\varphi_1}$, and $\text{PaPB}\models_{\mathbf{MP}_4}\underline{\varphi_2}$. We can prove their proofs are of length no greater than h. From the inductive hypothesis, $\mathsf{index}[\overline{AE_t},\overline{\varphi_1}] \le |\overline{AE_t}|$. However, we have $\mathsf{index}[\overline{AE_t},\overline{\varphi_1}] = \mathsf{index}[AE_t,\varphi_1]$. From the assumption $|AE_t| = \mathsf{index}[AE_t,\varphi_1]$. That is, $|AE_t| = \mathsf{index}[AE_t,\varphi_1] = \mathsf{index}[\overline{AE_t},\overline{\varphi_1}] \le |\overline{AE_t}|$. Thus, $|AE_t| \le |\overline{AE_t}|$. Since $\overline{AE_t} \subseteq AE_t$, $|AE_t| \not< |\overline{AE_t}|$. Hence $|AE_t| = |\overline{AE_t}|$. However, as $\overline{AE_t} \subseteq AE_t$, $\underline{AE_t} \subseteq AE_t$, $\underline{AE_t} \cap \overline{AE_t} \neq \emptyset$, and $\underline{AE_t} \neq \emptyset$, we obtain a contradiction. As a result, it holds that $\mathsf{index}[AE_t,\varphi_1] = \mathsf{index}[AE_t,\varphi_2] < |AE_t|$ for $2 \le t \le n$. The reason why SRT is not applied at the step $h+1$ is The reason why SRT is not applied is that q is a prov-query (i.e., $\mathsf{issuer}[\phi] = i \notin \mathsf{prov}[q]$). Thus the statement is true when the proof is of length $h+1$.

Proof Sketch of Theorem6.

(Claim 2 implies Claim 1.) We prove that all prov-aware queries satisfying $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{\{\mathsf{n}_1,\cdots,\mathsf{n}_l\}}\cdots\mathrm{D}_{\mathsf{n}_l}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$ also result in Claim 1. Note that l can be any natural number such that $1 \le l$. Since we prove for all the prov-aware queries that satisfies Claim 2, we also prove for all prov-aware queries of any l in Claim 2. We show that the Claim 2 implies Claim 1 by induction on the length of the proof for $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{\{\mathsf{n}_1,\cdots,\mathsf{n}_l\}}\cdots\mathrm{D}_{\mathsf{n}_l}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$. Assume the length of the proof is h.

Base case: $h = 2$. Since the proof consists only of two lines, it must hold that $l = 1$. Suppose that $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{\mathsf{n}_1}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$. We enumerate all the cases that give rise to $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{\mathsf{n}_1}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$. As PaPB is composed of formulas like $\mathrm{B}_i p_1$ and $\mathrm{T}^j_k p_2$, there are two possibilities how $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{\mathsf{n}_1}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$ can be concluded. The first possibility is to apply an instance of the axiom SRT: $\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p \Leftrightarrow \mathrm{D}_{\mathsf{n}}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$ to $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$. The other is to apply an instance of the axiom i-centric-Dlgt: $\mathrm{T}^{\mathsf{n}}_{\mathsf{n}_1} p \wedge \mathrm{T}^{\mathsf{n}_1}_{\mathsf{n}'} p \Rightarrow \mathrm{D}_{\mathsf{n}_1}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$ to $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}_1} p \wedge \mathrm{T}^{\mathsf{n}_1}_{\mathsf{n}'} p$. However, the fact $\mathsf{n} \neq \mathsf{n}_1$ rules out the first one. As a result, it is the second possibility that actually takes place. Since $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}_1} p \wedge \mathrm{T}^{\mathsf{n}_1}_{\mathsf{n}'} p$ appears in the first line of the proof, one can determine that $\{\mathrm{T}^{\mathsf{n}}_{\mathsf{n}_1} p, \mathrm{T}^{\mathsf{n}_1}_{\mathsf{n}'} p\} \subseteq \text{PaPB}$. Hence, from the definition of paths, Claim 1 holds.

Course-of-values inductive step. Assume that the proof is of length $h+1$, and that Claim 2 implies Claim 1 for all the proof of length less than h. We now show it is also the case for the proof of length $h+1$. We follow the same idea as in the base case. That is to enumerate all the cases that could lead to the conclusion $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{\{\mathsf{n}_1,\cdots,\mathsf{n}_l\}}\cdots\mathrm{D}_{\mathsf{n}_l}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$.

Case D3. The conclusion is reached by an application of an instance of the axiom D3: $\mathrm{D}_{AE_1}\varphi \Rightarrow \mathrm{D}_{AE_2}\varphi$ where $AE_1 \subset AE_2$. Suppose the instance is applied to $\text{PaPB}\models_{\mathbf{MP}_4} q_1$ to obtain $\text{PaPB}\models_{\mathbf{MP}_4} q_2$. If q_1 is prov-aware, then q_2 must be non-prov-aware. Assuming q_1 is non-prov-aware, q_2 can not be a prov-aware simply by an application of an instance of D3 to $\text{PaPB}\models_{\mathbf{MP}_4} q_1$. By Proposition 5, $\mathsf{index}[AE_1,q_1] \le |AE_1|$. In addition, $\mathsf{index}[AE_1,q_1] = \mathsf{index}[AE_2,q_2]$ and $AE_1 \subset AE_2$; hence, $\mathsf{index}[AE_2,q_2] < |AE_2|$. However,

Case AE-Red. Denote $\mathrm{D}_{\{\mathsf{n}_1,\cdots,\mathsf{n}_l\}}\cdots\mathrm{D}_{\{\mathsf{n}_t,\cdots,\mathsf{n}_l\}}\mathrm{D}_{\{\mathsf{n}_t,\cdots,\mathsf{n}_l\}}\cdots\mathrm{D}_{\mathsf{n}_l}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$ as φ. Suppose an instance of AE-Red: $\mathrm{D}_{\{\mathsf{n}_t,\cdots,\mathsf{n}_l\}}\mathrm{D}_{\{\mathsf{n}_t,\cdots,\mathsf{n}_l\}}\cdots\mathrm{D}_{\mathsf{n}_l}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p \Rightarrow \mathrm{D}_{\{\mathsf{n}_t,\cdots,\mathsf{n}_l\}}\cdots\mathrm{D}_{\mathsf{n}_l}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$ to $\text{PaPB}\models_{\mathbf{MP}_4}\varphi$ to obtain $\mathrm{D}_{\{\mathsf{n}_1,\cdots,\mathsf{n}_l\}}\cdots\mathrm{D}_{\{\mathsf{n}_t,\cdots,\mathsf{n}_l\}}\cdots\mathrm{D}_{\mathsf{n}_l}\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$. Since there are twice appearances of the set $\{\mathsf{n}_t,\cdots,\mathsf{n}_l\}$; we denote the former one as AE_{fo} and the latter one as AE_{la}. Since q is prov-aware, $\mathsf{index}[\{\mathsf{n}_t,\cdots,\mathsf{n}_l\},q] = t - l + 1$. And, $\mathsf{index}[AE_{fo},\varphi] = \mathsf{index}[AE_{la},\varphi] + 1 = \mathsf{index}[\{\mathsf{n}_t,\cdots,\mathsf{n}_l\},q] + 1 = t - l + 2 \ge |AE_{fo}|$. This is a contradiction to the Proposition 5. Hence, $\text{PaPB}\models_{\mathbf{MP}_4} q$ cannot be concluded by an application of the AE-Red.

Case i-centric-Dlgt. Suppose $\text{PaPB}\models_{\mathbf{MP}_4} q$ is obtained by an application of i-centric-Dlgt to $\text{PaPB}\models_{\mathbf{MP}_4}\mathrm{D}_{AE_t}\cdots\mathrm{D}_{AE_1}(\mathrm{T}^{\mathsf{n}}_{n_v} p \wedge \mathrm{T}^{n_v}_{\mathsf{n}'} p)$, for certain v such that $1 \le v \le l$. Then at some previous step in the proof we have

PaPB$\models_{\mathbf{MP_4}} \mathrm{D}_{AE_t} \cdots \mathrm{D}_{AE_1} \mathrm{T}^{\mathsf{n}}_{\mathsf{n}_v} p$. Denote $\mathrm{D}_{AE_t} \cdots \mathrm{D}_{AE_1} \mathrm{T}^{\mathsf{n}}_{\mathsf{n}_v} p$ as φ. By Proposition 3, we have PaPB$\models_{\mathbf{MP_4}} \mathrm{D}_{\overline{AE_t}} \cdots \mathrm{D}_{\overline{AE_1}} \mathrm{T}^{\mathsf{n}}_{\mathsf{n}_v} p$, where $\overline{AE_u} = AE_u \cap \mathsf{ag@path}[\mathsf{n}, \mathsf{n}_v]$, for $1 \leq u \leq t$. Denote $\mathrm{D}_{\overline{AE_t}} \cdots \mathrm{D}_{\overline{AE_1}} \mathrm{T}^{\mathsf{n}}_{\mathsf{n}_v} p$ as $\overline{\varphi}$. Note that $\mathsf{index}[AE_t, \varphi] = \mathsf{index}[\overline{AE_t}, \overline{\varphi}]$. By Proposition 5, we have $\mathsf{index}[\overline{AE_t}, \overline{\varphi}] \leq |\overline{AE_t}|$. Since $\overline{AE_t} \subseteq \mathsf{ag@path}[\mathsf{n}, \mathsf{n}_v]$, $|\overline{AE_t}| \leq v$ (note that $\mathsf{n}_v \notin \mathsf{ag@path}[\mathsf{n}, \mathsf{n}_v]$) . However, since q is prov-aware, $\mathsf{n} \notin AE_t$ and thus $\mathsf{n} \notin \overline{AE_t}$. So, $|\overline{AE_t}| \leq v - 1$. Hence $\mathsf{index}[AE_t, \varphi] \leq v - 1$.

Note that $\mathsf{index}[AE_t, q] = \mathsf{index}[AE_t, \varphi] + 1$. Therefore, $\mathsf{index}[AE_t, q] \leq v$. However, From the condition in Claim 2, $\mathsf{index}[\{\mathsf{n}_l, \cdots, \mathsf{n}_1\}, q] = l$. Thus, $v = l$. Hence, for q to be prov-aware, an instance of i-centric-Dlgt is applied to $\mathrm{D}_{\{\mathsf{n}_l, \cdots, \mathsf{n}_1\}} \cdots \mathrm{D}_{\{\mathsf{n}_l, \mathsf{n}_{l-1}\}} (\mathrm{T}^{\mathsf{n}}_{\mathsf{n}_l} p \wedge \mathrm{T}^{\mathsf{n}_l}_{\mathsf{n}'} p)$. And, at some steps prior to h,

$$\mathrm{D}_{\{\mathsf{n}_l, \cdots, \mathsf{n}_1\}} \cdots \mathrm{D}_{\{\mathsf{n}_l, \mathsf{n}_{l-1}\}} \mathrm{T}^{\mathsf{n}}_{\mathsf{n}_l} p \tag{24}$$

$$\mathrm{D}_{\{\mathsf{n}_l, \cdots, \mathsf{n}_1\}} \cdots \mathrm{D}_{\{\mathsf{n}_l, \mathsf{n}_{l-1}\}} \mathrm{T}^{\mathsf{n}_l}_{\mathsf{n}'} p \tag{25}$$

By Proposition 3, from (24), we have $\mathrm{D}_{\{\mathsf{n}_{l-1}, \cdots, \mathsf{n}_1\}} \cdots \mathrm{D}_{\mathsf{n}_{l-1}} \mathrm{T}^{\mathsf{n}}_{\mathsf{n}_l} p$. Note the length of this proof is less than h. This is because $\mathsf{n}_l \notin \mathsf{ag@path}[\mathsf{n}, \mathsf{n}_l]$. From the inductive hypothesis, $\mathsf{DS}\langle$PaPB$\rangle$ has a path $\{\langle \mathsf{n}, \mathsf{n}_1\rangle, \langle \mathsf{n}_1, \mathsf{n}_2\rangle, \cdots, \langle \mathsf{n}_{l-1}, \mathsf{n}_l\rangle\}$ associated with p. Also by Proposition 3, from (25), we have $\mathrm{D}_{\{\mathsf{n}_l\}} \cdots \mathrm{D}_{\{\mathsf{n}_l\}} \mathrm{T}^{\mathsf{n}_l}_{\mathsf{n}'} p$, because $\mathsf{ag@path}[\mathsf{n}_l, \mathsf{n}'] = \{\mathsf{n}_l\}$. Suppose otherwise that $\mathsf{n}_u \in \mathsf{ag@path}[\mathsf{n}_l, \mathsf{n}']$ for certain u such that $1 \leq u \leq l - 1$. Then there is a path from n_l to n_u associated with p. However, $\mathsf{DS}\langle$PaPB$\rangle$ has a path $\{\langle \mathsf{n}, \mathsf{n}_1\rangle, \langle \mathsf{n}_1, \mathsf{n}_2\rangle, \cdots, \langle \mathsf{n}_{l-1}, \mathsf{n}_l\rangle\}$ associated with p; a contradiction with the fact the PaPB is acyclic. Therefore, by repeating instances of the axiom SRT: $\mathrm{D}_{\mathsf{n}_l} \mathrm{T}^{\mathsf{n}_l}_{\mathsf{n}'} p \Leftrightarrow \mathrm{T}^{\mathsf{n}_l}_{\mathsf{n}'} p$, we have $\mathrm{T}^{\mathsf{n}_l}_{\mathsf{n}'} p$. According to the definition of paths, one can conclude that $\mathsf{DS}\langle$PaPB$\rangle$ has a path $\{\langle \mathsf{n}, \mathsf{n}_1\rangle, \langle \mathsf{n}_1, \mathsf{n}_2\rangle, \cdots, \langle \mathsf{n}_l, \mathsf{n}'\rangle\}$ associated with p.

(Claim 3 implies Claim 2.) Denote $\mathrm{D}_{\{\mathsf{n}_1, \cdots, \mathsf{n}_l, \mathsf{n}'\}} \cdots \mathrm{D}_{\{\mathsf{n}_l, \mathsf{n}'\}} \mathrm{D}_{\mathsf{n}'} \mathrm{B}_{\mathsf{n}} p$ as q. Suppose that PaPB$\models_{\mathbf{MP_4}} q$ is obtained by an application of the axiom Dlgt to PaPB$\models_{\mathbf{MP_4}} \mathrm{D}_{AE_t} \cdots \mathrm{D}_{AE_1} (\mathrm{T}^{\mathsf{n}}_{\mathsf{n}_v} p \wedge \mathrm{B}_{\mathsf{n}_v} p)$. Then PaPB$\models_{\mathbf{MP_4}} \mathrm{D}_{AE_t} \cdots \mathrm{D}_{AE_1} \mathrm{T}^{\mathsf{n}}_{\mathsf{n}_v} p$ holds at some previous step. Let φ denote $\mathrm{D}_{AE_t} \cdots \mathrm{D}_{AE_1} \mathrm{T}^{\mathsf{n}}_{\mathsf{n}_v} p$. By Proposition 3, there is a path from n to n_v associated with p. By Proposition 5, $\mathsf{index}[AE_t, \varphi] \leq |AE_t| = |\mathsf{ag@path}[\mathsf{n}, \mathsf{n}_v]| - 1$ (for $\{\mathsf{n}\} \not\subset AE_t$ as q is a prov-query.). Since $\mathsf{index}[\{\mathsf{n}_1 \cdots, \mathsf{n}_l, \mathsf{n}'\}, q] = l + 1$ and $\mathsf{index}[\{\mathsf{n}_1 \cdots, \mathsf{n}_l, \mathsf{n}'\}, q] = \mathsf{index}[AE_t, \varphi] + 1$, it holds that $\mathsf{index}[AE_t, \varphi] = l$. Hence, $l \leq |\mathsf{ag@path}[\mathsf{n}, \mathsf{n}_v]| - 1$, i.e., $|\mathsf{ag@path}[\mathsf{n}, \mathsf{n}_v]| \geq l + 1$. Since $|\mathsf{ag@path}[\mathsf{n}, \mathsf{n}_v]| \not> l + 1$ (for there are $l + 2$ nodes all together and the ending node of a path does not belong to the set of agents in the path.), it follows that $|\mathsf{ag@path}[\mathsf{n}, \mathsf{n}_v]| = l + 1$. Therefore, n_v can only be n'.

To obtain PaPB$\models_{\mathbf{MP_4}} q$ by applying the instance of the axiom Dlgt: $\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p \wedge \mathrm{B}_{\mathsf{n}'} p \Rightarrow \mathrm{D}_{\mathsf{n}'} \mathrm{B}_{\mathsf{n}} p$ to PaPB$\models_{\mathbf{MP_4}} \mathrm{D}_{AE_t} \cdots \mathrm{D}_{AE_1} (\mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p \wedge \mathrm{B}_{\mathsf{n}'} p)$, φ must be $\mathrm{D}_{\{\mathsf{n}_1, \cdots, \mathsf{n}_l, \mathsf{n}'\}} \cdots \mathrm{D}_{\mathsf{n}_l, \mathsf{n}'} \mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$. Thus PaPB$\models_{\mathbf{MP_4}} \mathrm{D}_{\{\mathsf{n}_1, \cdots, \mathsf{n}_l, \mathsf{n}'\}} \cdots \mathrm{D}_{\{\mathsf{n}_l, \mathsf{n}'\}} \mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$. By Proposition 3, we have PaPB$\models_{\mathbf{MP_4}} \mathrm{D}_{\{\mathsf{n}_1, \cdots, \mathsf{n}_l\}} \cdots \mathrm{D}_{\{\mathsf{n}_l\}} \mathrm{T}^{\mathsf{n}}_{\mathsf{n}'} p$, because $\mathsf{n}' \notin \mathsf{ag@path}[\mathsf{n}, \mathsf{n}']$.

(Claim 1 implies Claim 3.) Obvious according to the definitions and the axioms.

Proof Sketch of Theorem 7.

It is obvious that if ePaPB$'\models_{\mathbf{MP_4}} q$ then ePaPB$\models_{\mathbf{MP_4}} q$. We now prove that if ePaPB$\models_{\mathbf{MP_4}} q$ then ePaPB$'\models_{\mathbf{MP_4}} q$. Suppose, for the sake of contradiction, that ePaPB$\models_{\mathbf{MP_4}} q$ but ePaPB$'\not\models_{\mathbf{MP_4}} q$. Then there exists a set MSP $\subseteq$ ePaPB such that MSP$\models_{\mathbf{MP_4}} q$ and for any MSP$' \subset$ MSP it holds that MSP$'\not\models_{\mathbf{MP_4}} q$. MSP is short for Maximal Subset of Policies. We enumerate the differences that give rise to MSP$\models_{\mathbf{MP_4}} q$ but ePaPB$'\not\models_{\mathbf{MP_4}} q$.

(1) Suppose that there exists certain rules $(c_1 \wedge \cdots \wedge c_n \Rightarrow h) \in$ MSP but $h \notin$ ePaPB$'$ (i.e., $h \notin \mathsf{activehead}$(ePaPB)). Since the proof of MSP$\models_{\mathbf{MP_4}} q$ makes use of the rule $(c_1 \wedge \cdots \wedge c_n \Rightarrow h)$, MSP$\models_{\mathbf{MP_4}} h$. Otherwise, according to the structure of q, this rule would not be used in the proof. However, as $h \notin \mathsf{activehead}$(ePaPB), ePaPB$\not\models_{\mathbf{MP_4}} h$; a contradiction. (2) Suppose that MSP$\models_{\mathbf{MP_4}} \psi$ where ψ of the form $\mathrm{B}_i \mathrm{B}_j p$ or $\mathrm{B}_i \mathrm{T}^j_k p'$. If ψ is only used to derive h from the rule $(c_1 \wedge \cdots \wedge c_n \Rightarrow h)$, then ePaPB$'$ should contain h for $h \in \mathsf{activehead}$(ePaPB); and, as a result, ePaPB$'\models_{\mathbf{MP_4}} q$, a contradiction. Otherwise, an application of axioms is applied to ψ. However, the formulas derived from ψ by axioms and rules of inference are of the forms $\mathrm{B}_i \psi_1 \wedge \psi_2$ or $\mathrm{B}_i \mathrm{D}_{AE_n} \cdots \mathrm{D}_{AE_1} \phi$. According to the structure of q and the construction of ePaPB, ψ would not be used in the proof. Then ePaPB$'\models_{\mathbf{MP_4}} q$ should have held, a contradiction. (3) Suppose that $p_0 \in$ MSP where $p_0 \in$ Prop. In compliance with the structure of q and the construction of ePaPB, p_0 can only be used to derive h from the rule $(c_1 \wedge \cdots \wedge c_n \Rightarrow h)$. In this case, however, ePaPB$'$ should contain h for $h \in \mathsf{activehead}$(ePaPB); a contradiction would be reached.

Therefore, if MSP$\models_{\mathbf{MP_4}} q$ then ePaPB$'\models_{\mathbf{MP_4}} q$. From the definition of MSP, it follows that if ePaPB$\models_{\mathbf{MP_4}} q$ then ePaPB$'\models_{\mathbf{MP_4}} q$.