Managing Authorization Provenance: A Modal Logic based Approach

Jinwei Hu*†, Yan Zhang†, Ruixuan Li* and 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
Email: jwhu@hust.edu.cn, yan@scm.uws.edu.au, {rxli, zdlu}@hust.edu.cn

***Abstract*—In distributed environments, access control decisions depend on statements of multiple agents rather than only one central trusted party. However, existing policy languages put few emphasis on *authorization provenances*. The capability of managing these provenances is important and useful in various security areas such as computer auditing and safeguarding delegations. Based on the newly proposed logic, we define one type of authorization provenances. We exemplify the applications of these provenances by a case study.**

I. Introduction

Recently, major research efforts have applied logics into the design of policy languages to deal with distributed authorizations [1], [2], [4], [5]. The set of policies written in a policy language is regarded as a *policy base*. When a principal requests resources, the request is translated to a query of the policy base. Then the access is granted if the answer to the query is positive and denied otherwise.

Existing access control systems based on previous policy languages, however, failed to support the *management of authorization provenances*. Informally, an authorization provenance denotes the set of agents whose statements are referenced in the deduction of an authorization decision. In traditional centralized authorizations, a central trusted party makes authorization decisions and takes the responsibility 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, a set of agents besides the central party (e.g., delegatees) may play a role in and be responsible for access control decision making.

There are several reasons why it is important for one to manage authorization provenances. First, host security may be compromised if provenances are not taken into account when making authorization decisions [7], [9]. Wang et al., [9] found that users may abuse delegations to circumvent security policies; and proposed a defending mechanism, source-based enforcement, which checks not only if a subject has a privilege but also who, if any, delegated this privilege to the subject. Again, In [7], authors pointed out that, since existing enforcement of *Discretionary Access Control* (DAC) models cannot correctly identify the true origins of a request, they fail to defense against trojan horses and buggy programs. To trace the identity of requesters and thus protect against these attacks, the authors then invented a model based on a notion of *a contamination source*. It is worth noting that both the reasons and defense mechanisms of these security breaches are closely related to authorization provenances.

On the other hand, auditing is an indispensable part of a secure system. One objective of auditing is to identify from where security breaches started. There arises a trend to include proofs of authorization decisions in system logs for auditing [8]. Armed with the ability to reason about authorization provenances, one may make more use of logs. For example, since provenances record the agents involved, they can help trace back to the origins of security compromises.

From the above observations, we attempted to design an authorization logic, named DBT, which treats provenances explicitly [3]. DBT builds upon a logic BT [6]. The BT logic can represent belief and trust (delegation) and their relations. 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. Based on DBT, we define a notion of authorization provenances. To the best of our knowledge, this work is the first to define provenances for authorizations in logic.

II. The access control logic DBT

A. Syntax

Consider a 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 and $\mathrm{T}^i_j\varphi$ reads that agent i trusts agent j on φ. $\mathrm{D}_i\varphi$ means that "due to agent i, φ holds" or that i causes that φ holds. Given an *agent expression* $AE \subseteq \mathcal{AG}$, 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 cause φ. Let Prop be a set of primitive propositions. The set WFF of well-formed formulas (wff) is defined as follows:

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

A *policy base PB* is a finite subset of WFF. We refer to the agent who enforces access control policies in the

system in question as Local. Local is the root of trust which protects the requested resources, assembles the policy base, and evaluates queries to make access control decisions.

B. Semantics

We describe a semantics of DBT based on Krpike 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

- W is a set of possible worlds,
- $\pi : W \mapsto 2^{\mathsf{Prop}}$ is a labeling function which maps each world to a subset P of Prop such that any $p \in P$ is true in this world and any $p \in \mathsf{Prop} \backslash P$ is false in this world,
- $\mathcal{B}_i \subseteq W \times W$ is a serial, transitive and Euclidean binary relation on W,
- $\mathcal{D}_i \subseteq W \times W$ is a binary relation on W, and
- $\mathcal{T}^i_j \subseteq W \times 2^W$ 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 $\models$ for the nontrivial cases: $\mathrm{B}_i\varphi$, $\mathrm{D}_i\varphi$, $\mathrm{D}_{AE}\varphi$, and $\mathrm{T}^i_j\varphi$.

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

C. The axiomatic system of DBT

To capture the properties of distributed access control, we make the following constraints on the models, and call the class of models satisfying these constraints as *models for access control*, written **MAC**.

C1 for all $S \in \mathcal{T}^i_j(w)$, if $\mathcal{B}_j(w) \subseteq S$, then $\mathcal{D}_j \circ \mathcal{B}_i(w) \subseteq S$, [1]
C2 $\mathcal{D}_i(w) \subseteq \mathcal{D}_i \circ \mathcal{D}_i(w)$,
C3 $\mathcal{T}^i_j(w) = \bigcap_{u \in \mathcal{B}_i(w)} \mathcal{T}^i_j(u)$,
C4 $\mathcal{T}^i_j(w) = \bigcap_{u \in \mathcal{D}_i(w)} \mathcal{T}^i_j(u)$,
C5 $\mathcal{B}_i(w) = \mathcal{D}_i \circ \mathcal{B}_i(w)$,

[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 $R \circ Q = \{(x, z) \mid \exists y \in \mathcal{Y} : \ y \in R(x) \wedge z \in Q(y)\}$.

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 $AE = \{i\}$, $i \in \mathcal{AG}$
DBT1 (delegation): $\mathrm{T}^i_j\varphi \wedge \mathrm{B}_j\varphi \Rightarrow \mathrm{D}_j\mathrm{B}_i\varphi$
DBT2 (reduction): $\mathrm{D}_i\mathrm{D}_i\varphi \Rightarrow \mathrm{D}_i\varphi$
DBT3 (self aware delegation): $\mathrm{B}_i\mathrm{T}^i_j\varphi \Leftrightarrow \mathrm{T}^i_j\varphi$
DBT4 (self responsible delegation): $\mathrm{D}_i\mathrm{T}^i_j\varphi \Leftrightarrow \mathrm{T}^i_j\varphi$
DBT5 (self responsible belief): $\mathrm{D}_i\mathrm{B}_i\varphi \Leftrightarrow \mathrm{B}_i\varphi$
DBT6 (i-centric delegation): $\mathrm{T}^i_j\varphi \wedge \mathrm{T}^j_k\varphi \Rightarrow \mathrm{D}_j\mathrm{T}^i_k\varphi$
DBT7 (AE-reduction): $\mathrm{D}_{AE}\mathrm{D}_{AE}\varphi \Rightarrow \mathrm{D}_{AE}\varphi$
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 *AC*

C6 $\mathcal{T}^i_j(w) \cap \mathcal{T}^j_k(w) \subseteq \bigcap_{u \in \mathcal{D}_j(w)} \mathcal{T}^i_k(u)$, and,
C7 $\mathcal{D}_{AE}(w) \subseteq \mathcal{D}_{AE} \circ \mathcal{D}_{AE}(w)$.

We proposed an axiomatic system *AC* in Figure 1. Constraints C1 to C7 correspond to axioms DBT1 to DBT7, respectively.

Theorem 1: The axiomatic system *AC* is sound and complete with respect to **MAC**. [3]

A *query* is a WFF formula. We say a policy base *PB entails* a query q, written $PB \models_{MAC} q$, if and only if, for all $\mathcal{M} \in$ **MAC** and states w in $\mathcal{M}$, if for all $\psi \in PB$ $\langle \mathcal{M}, w \rangle \models \psi$ then $\langle \mathcal{M}, w \rangle \models q$.

III. Authorization Provenances

The basic requirement for managing authorization provenances is to prevent provenance loss and forging. However, policy bases constructed by DBT as a whole do not come up to this standard. Because one is free to write policies like $\mathrm{D}_{AE}\varphi_1 \Rightarrow \psi_1$ and $\varphi_2 \Rightarrow \mathrm{D}_{AE}\psi_2$. In the first case the provenance information is lost when ψ_1 is derived, whereas unreal provenances may be forged in the second case when $\mathrm{D}_{AE}\psi_2$ is concluded.

Now we identify a subset of DBT which is provenance-aware in the sense that a class of queries which incorporate the provenance information in themselves can be evaluated against this subset correctly.

A. Definitions of WFF_{AP}

We denote the set of η formulas satisfying the following syntax as WFF_{AP}. Given $p \in \mathsf{Prop}$,

$$\phi ::= \mathrm{T}^i_j p \mid \mathrm{B}_i p \mid \mathrm{D}_{AE}\phi$$
$$\eta ::= \phi \mid \phi_1 \wedge \cdots \wedge \phi_n \Rightarrow \phi \quad (n \geq 0)$$

Since policy languages in the literature [2], [5] have taken similar forms as WFF_{AP}, policy bases written in WFF_{AP} is expressive enough for authorizations. Unless otherwise stated, when referring to a policy base we mean a policy base specified using WFF_{AP}.

We say that φ is a *subformula* of η if either (a) $\varphi = \eta$, (b) η is of the form $\neg\eta'$, $\mathrm{B}_i\eta'$, $\mathrm{D}_{AE}\eta'$, $\mathrm{T}^i_j\eta'$, and φ is a subformula of η', or (c) η is of the form $\eta_1 \wedge \cdots \wedge \eta_n \Rightarrow \eta_0$ and φ is a subformula of one of η_l $(0 \leq l \leq n)$. We denote the set of subformulae of η as $Sub(\eta)$.

Since provenances concern the agents whose statements contribute to the derivation of conclusions, we abstract agents from each formula in policy bases. On the other hand, we also need to extract the authorization-related contents from each formula. Thus, we define two mappings, U and CC, on the structures of $\eta \in \mathsf{WFF}_{AP}$.

- The mapping $U : \mathsf{WFF}_{AP} \mapsto \mathcal{AG}$ is defined as:
  1) $U(\mathrm{B}_i\varphi) = U(\mathrm{T}^i_j\varphi) = i$,
  2) $U(\mathrm{D}_{AE}\phi) = U(\phi)$, and
  3) $U(\phi_1 \wedge \cdots \wedge \phi_n \Rightarrow \phi) = U(\phi)$.
- The mapping $CC : \mathsf{WFF}_{AP} \mapsto \mathsf{WFF}_{AP}$ is defined as:
  1) if η is a B-formula or a T-formula, then $CC(\eta) = \eta$,
  2) if η is a D-formula of the form $\mathrm{D}_{AE}\phi$, then $CC(\eta) = CC(\phi)$, and
  3) otherwise, $CC(\eta) = \eta$.

Example 1: $\mathrm{T}^{\mathsf{Alice}}_{\mathsf{Bob}}\mathsf{goodPeer}(\mathsf{David})$ is from the agent Alice, namely $U(\mathrm{T}^{\mathsf{Alice}}_{\mathsf{Bob}}\mathsf{goodPeer}(\mathsf{David})) = \mathsf{Alice}$; and Bob issues the statement $\mathrm{B}_{\mathsf{Bob}}\mathsf{goodPeer}(\mathsf{David})$, thus $U(\mathrm{B}_{\mathsf{Bob}}\mathsf{goodPeer}(\mathsf{David})) = \mathsf{Bob}$. Though $\mathrm{D}_{\mathsf{Bob}}\mathrm{B}_{\mathsf{Alice}}\mathsf{goodPeer}(\mathsf{David})$ means that Bob causes Alice to believe goodPeer(David), Alice is still responsible for, if any, conclusions derived from this formula. Therefore, $U(\mathrm{D}_{\mathsf{Bob}}\mathrm{B}_{\mathsf{Alice}}\mathsf{goodPeer}(\mathsf{David})) = \mathsf{Alice}$. And for this formula, the authorization-related content is that Alice believes goodPeer(David); namely, $CC(\mathrm{D}_{\mathsf{Bob}}\mathrm{B}_{\mathsf{Alice}}\mathsf{goodPeer}(\mathsf{David})) = \mathrm{B}_{\mathsf{Alice}}\mathsf{goodPeer}(\mathsf{David})$.

Definition 2 (Trace): Given a formula η, we define the *trace* of η, written $Tr(\eta)$, on the structure of η:

1) if η is a D-formula of the form $\mathrm{D}_{AE_1}\cdots\mathrm{D}_{AE_n}\varphi$ where φ is *not* a D-formula, $Tr(\eta) = AE_1 \parallel \cdots \parallel AE_n$,
2) otherwise, $Tr(\eta) = \emptyset$.

Basically, traces capture the agents whose statements are used in the reasoning process of a formula. If curious about from where a belief is concluded, one can query q with $CC(q)$ being the belief but with variable $Tr(q)$.

Example 2: Consider $q_1 = \mathrm{D}_{\mathsf{Bob}}\mathrm{B}_{\mathsf{Alice}}\mathsf{goodPeer}(\mathsf{David})$, $Tr(q_1) = \mathsf{Bob}$; if querying q_1 against the policy base, one is asking if it is Bob who causes $\mathrm{B}_{\mathsf{Alice}}\mathsf{goodPeer}(\mathsf{David})$ to be concluded. One can also ask if Alice has the belief that goodPeer(David) because of Cathy by the query $\mathrm{D}_{\mathsf{Cathy}}\mathrm{B}_{\mathsf{Alice}}\mathsf{goodPeer}(\mathsf{David})$. For any B-formula or T-formula, say $q_2 = \mathrm{B}_{\mathsf{Alice}}\mathsf{goodPeer}(\mathsf{David})$, $Tr(q_2) = \emptyset$ because q_2 denotes that $\mathrm{B}_{\mathsf{Alice}}\mathsf{goodPeer}(\mathsf{David})$ holds just because of $U(q_2) = \mathsf{Alice}$ herself but no agent else.

Supposing a trace $Tr(q) = AE_1 \parallel \cdots \parallel AE_n$, we define the set of agents that appear in $Tr(q)$ as $AgentsIn(Tr(q)) = \{i \in \mathcal{AG} \mid$ there exists $1 \leq l \leq n$ such that $i \in AE_l\}$. When AE is a singleton set, say $\{i\}$, we write i to denote AE.

B. Authorization-provenance-aware (AP) policy bases and queries

Given a policy base PB, we say PB is an AP policy base if, for each $\eta \in$ PB, **SR1**, **SR2**, and **SR3** are satisfied.

SR1: If $\mathrm{D}_{AE}\mathrm{B}_i\varphi \in Sub(\eta)$ or $\mathrm{D}_{AE}\mathrm{T}^i_j\varphi \in Sub(\eta)$, then $i \equiv$ Local.

SR1 points out that AP policy bases are Local-centric in the sense that all conclusions from delegations are related to Local. In other words, authorization provenances are managed by the agent Local. This is intuitive because Local is the security guard and resources are granted only if somehow Local believes this authorization is legal.

SR2: η is *not* a D-formula.

SR2 simply prevents traces from being forged. If an agent Alice can issue that $\eta' = \mathrm{D}_{AE}\mathrm{B}_{\mathsf{Alice}}\varphi$, Alice is able to forge arbitrary traces AE for $\mathrm{B}_{\mathsf{Alice}}\varphi$; then traces of conclusions, which are derived from η', are unreal as a result.

SR3: If η is of the form $\phi_1 \wedge \cdots \wedge \phi_n \Rightarrow \phi$, then

A $U(\phi) = U(\phi_l)$, for $1 \leq l \leq n$.

B for any $1 \leq l \leq n$ if $i \in AgentsIn(Tr(\phi_l))$ then $i \in AgentsIn(Tr(\phi))$.

In **SR3**, item **A** is met by many policy languages in literature such as [2], [5]; so it does not restrict expressiveness as to security policy specifications. Item **B** simply records provenances.

Definition 3 (AP queries): An AP query q is a formula such that (1) $q \in \mathsf{WFF}_{AP}$, (2) taking form of $\mathrm{B}_i\phi$, $\mathrm{D}_{AE}\phi$, or $\mathrm{T}^i_j\phi$, (3) $U(q) =$ Local and (4) $Tr(q) \subseteq \mathcal{AG}$.

For example, $q_1 = \mathrm{B}_{\mathrm{Local}}\varphi_1$ and $q_2 = \mathrm{D}_{\{\mathsf{Alice},\mathsf{Bob}\}}\mathrm{B}_{\mathrm{Local}}\varphi_2$ are AP queries, whereas $q_3 = \mathrm{D}_{\mathsf{Alice}}\mathrm{B}_{\mathsf{Cathy}}\varphi_3$ and $q_4 = \mathrm{D}_{\mathsf{Alice}}\mathrm{D}_{\mathsf{Bob}}\mathrm{B}_{\mathrm{Local}}\varphi_2$ are not.

Proposition 2: Given an AP PBand an AP query q,

$$\mathrm{PB} \models_{MAC} q \text{ if and only if } \mathrm{PB}' \models_{MAC} q,$$

where PB′ is a subset of PB such that for all $\eta \in$ PB, $U(\eta) \in AgentsIn(Tr(q)) \cup \{U(q)\}$ if and only if $\eta \in$ PB′.

Proposition 2 shows the basic motivation of AP policy bases and queries. Accordingly, for an AP query q, $Tr(q)$ includes the agents whose statement are referenced in the deduction of q.

C. A Case Study

We use an example from [9] to illustrate the motivations of AP policy bases.

> 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. Alice is a Treasurer and Bob is a Clerk of the company. They decided to collude to issue checks for themselves.

As noted in [9], Alice and Bob are able to issue checks for themselves, through the following actions: (A1) Alice delegates the role Treasurer to Bob; (A2) Bob performs *pre* to prepare a check for Alice; and (A3) Alice performs *app* to approve the check prepared by Bob.

One may formally represent the scenario as follows.

$$\mathrm{B}_{\text{Local}}(\texttt{InRole}(\mathsf{Bob},\mathsf{Cl}) \wedge \texttt{InRole}(\mathsf{Alice},\mathsf{Tr})) \Rightarrow \mathrm{T}^{\text{Local}}_{\mathsf{Alice}}\texttt{InRole}(\mathsf{Bob},\mathsf{Tr}) \quad (1)$$

$$\mathrm{D}_{\mathsf{Alice}}\mathrm{B}_{\text{Local}}\texttt{InRole}(\mathsf{Bob},\mathsf{Tr}) \Rightarrow \mathrm{D}_{\mathsf{Alice}}\mathrm{T}^{\text{Local}}_{\mathsf{Bob}}\texttt{pre}(\mathit{check}) \quad (2)$$

$$\mathrm{B}_{\text{Local}}\texttt{InRole}(\mathsf{Alice},\mathsf{Tr}) \Rightarrow \mathrm{T}^{\text{Local}}_{\mathsf{Alice}}\texttt{app}(\mathit{check}) \quad (3)$$

From the assumption that Alice is a Treasurer and Bob is a Clerk and that (1), (4) holds. With action (A1), Bob brings a credential (5) which, together with (4), derives (6) by the axiom DBT1. From the implication (2), we have (7). With action (A2), it holds that $\mathrm{B}_{\mathsf{Bob}}\texttt{pre}(\mathit{check})$. Then by (self responsible belief) axiom DBT5, we have $\mathrm{D}_{\mathsf{Bob}}\mathrm{B}_{\mathsf{Bob}}\texttt{pre}(\mathit{check})$; further by axiom D3 it follows that (8). Again, by the axiom D3 it follows from (7) that (9) holds. Then, from (8) and (9), it follows that (10) by applying the axioms D2, DBT1, D3, and DBT7 in sequence. With action (A3), it holds that $\mathrm{B}_{\mathsf{Alice}}\texttt{app}(\mathit{check})$. Likewise, from (3), one can reach that (11).

$$\mathrm{T}^{\text{Local}}_{\mathsf{Alice}}\texttt{InRole}(\mathsf{Bob},\mathsf{Tr}) \quad (4)$$

$$\mathrm{B}_{\mathsf{Alice}}\texttt{InRole}(\mathsf{Bob},\mathsf{Tr}) \quad (5)$$

$$\mathrm{D}_{\mathsf{Alice}}\mathrm{B}_{\text{Local}}\texttt{InRole}(\mathsf{Bob},\mathsf{Tr}) \quad (6)$$

$$\mathrm{D}_{\mathsf{Alice}}\mathrm{T}^{\text{Local}}_{\mathsf{Bob}}\texttt{pre}(\mathit{check}) \quad (7)$$

$$\mathrm{D}_{\{\mathsf{Alice},\mathsf{Bob}\}}\mathrm{B}_{\mathsf{Bob}}\texttt{pre}(\mathit{check}) \quad (8)$$

$$\mathrm{D}_{\{\mathsf{Alice},\mathsf{Bob}\}}\mathrm{T}^{\text{Local}}_{\mathsf{Bob}}\texttt{pre}(\mathit{check}) \quad (9)$$

$$\mathrm{D}_{\{\mathsf{Alice},\mathsf{Bob}\}}\mathrm{B}_{\text{Local}}\texttt{pre}(\mathit{check}) \quad (10)$$

$$\mathrm{D}_{\mathsf{Alice}}\mathrm{B}_{\text{Local}}\texttt{app}(\mathit{check}) \quad (11)$$

Then one may query $q_5 = \mathrm{D}_{\mathsf{Alice}}\mathrm{B}_{\text{Local}}\texttt{pre}(\mathit{check})$, $q_6 = \mathrm{D}_{\mathsf{Bob}}\mathrm{B}_{\text{Local}}\texttt{pre}(\mathit{check})$, $q_7 = \mathrm{D}_{\{\mathsf{Alice},\mathsf{Bob}\}}\mathrm{B}_{\text{Local}}\texttt{pre}(\mathit{check})$, and $q_8 = \mathrm{D}_{\mathsf{Alice}}\mathrm{B}_{\text{Local}}\texttt{app}(\mathit{check})$. From above reasonings, we have PB $\not\models_{MAC} q_5$, PB $\not\models_{MAC} q_6$, but PB $\models_{MAC} q_7$ and PB $\models_{MAC} q_8$. Hence, one is informed that $\mathrm{B}_{\text{Local}}\texttt{pre}(\mathit{check})$ (i.e., *check* is authorized to be prepared) is the result of statements of Alice and Bob together, but not any individual one, and that, in contrast, $\mathrm{B}_{\text{Local}}\texttt{app}(\mathit{check})$ (i.e., *check* is authorized to be approved) can be ascribed to merely Alice.

IV. Concluding Remarks

We define the notion of authorization provenances in DBT logic, and show its usefulness through a case study. However, authorization provenances can be more complex when, for example, *attribute-based delegations* are allowed. We have defined a notion of *strong authorization provenance* (SAP) accordingly to capture those situations. Space limitations preclude the presentation of strong authorization provenance; Readers are referred to [3].

As future work, we are in the process of formalizing the transformation from a policy base to an SAP one and related properties. In addition, we are working on a type of SAP queries whose traces encode their provenances.

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, Open Foundation of State Key Laboratory of Software Engineering under Grant SKLSE20080718. This project is supported in part by an Australian Research Council (ARC) Discovery Projects Grant (DP0988396).

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] M. Y. Becker, C. Fournet, and A. D. Gordon. Design and semantics of a decentralized authorization language. In *20th IEEE Computer Security Foundations Symposium*, pages 3–15, 2007.

[3] J. Hu, Y. Zhang, and R. Li. A logic for authorization provenance. Technical report, Huazhong University of Science and Technology & University of Western Sydney, 2009.

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

[5] 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.

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

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

[8] J. A. Vaughan, L. Jia, K. Mazurak, and S. Zdancewic. Evidence-based audit. In *21st IEEE Computer Security Foundations Symposium*, pages 177–191, 2008.

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