Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 653 – 657

Procedia
Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)

Algebraic Model for Handling Access Control Policies

Khair Eddin Sabri[a,*], Hazem Hiary[a,*]

[a] *Department of Computer Science, King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan*

Abstract

Confidentiality of information is an important aspect that developers should take into consideration when building systems. One way to achieve confidentiality is to define access control policies that give authorization rules for allowing users to access resources. In large organizations, managing policies becomes a complex task. Usually, based on the defined policies, developers would need to manipulate policies such as composing them and enforcing predefined security constraints. In this paper, we present an algebraic model for specifying access control policies. It consists of a few number of operators which gives simplicity in specifying policies. The proposed model enables us to specify policies and enforce predefined security constraints. Furthermore, the model allows us to combine policies and analyze their effect on predefined constraints. Furthermore, it enables comparing the sensitivity of objects (e.g. files) and authority of subjects (e.g. users).

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license
(http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of the Conference Program Chairs

Keywords: Information Security, Access Control Policies, Formal Specification, Algebraic Analysis

1. Introduction

Many organizations require that their information be stored securely so that it can be accessed only by legitimate users while others should be denied this access. One way to provide the confidentiality of information is to control its access and limit it to legitimate users only. Therefore, policies are specified to describe the allowed access for each user, and mechanisms are implemented to enforce those policies. In large organizations where several policies are defined, it becomes more complicated to manage policies. For example, an administrator may need to ensure safety properties by enforcing predefined constraints. A policy can be seen as a rule which states a privilege of a user to access a specific object, while constraints are expressions defined on policies. For example, a constraint can state that a user should not be able to access two objects belonging to the same conflict class.

In this paper, we propose an algebraic model to analyze access control policies. Our model is information algebra[1] which is an abstract algebraic system that links several representations of information into one structure. The motivation of using information algebra is to define our theory within a setting that is well studied and have an enrich theory. Furthermore, it enables us to connect different representations of access control policies as all policies are information

* Corresponding author. Tel.: +962-6-5355-000 ext. 22557 ; fax: +962-6-5355-522.
E-mail address: k.sabri@ju.edu.jo hazemh@ju.edu.jo

1877-0509 © 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license
(http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.146

and therefore, can be represented within information algebra. This paper introduces an algebraic model to specify policies. The model handles policies and set of policies. It allows enforcing constraints and comparing policies as well as set of policies. Furthermore, it allows from a set of policies to extract and compare the sensitivity of objects and authority of users.

The structure of the paper is as follows: Section 2 summarizes information algebra. Section 3 introduces the algebraic model. Section 4 presents the uses of the model. Section 5 presents related works. Finally, we conclude in Section 6.

2. Information algebra

Kohlas and Stärk[1] introduced information algebra which an abstract algebraic structure $((\Phi,\cdot),(D,\curlyvee,\curlywedge),\downarrow,d,e)$ to connect several representations of information such as algebraic specifications, relational databases, modules, and constraint systems. It is applied to data cleaning[2] and used to analyze information flow[3]. Information algebra contains a set of information Φ and an operator $\cdot$ to represent combining information. Also, it contains a lattice of frames $(D,\curlyvee,\curlywedge)$ with an ordering relation $\preccurlyeq$ that gives a classification to the information. Furthermore, information algebra contains an operator $\downarrow$ to restrict an information to a specific frame, and an operator d that gives the frame of an information. Each frame x contains an empty information e_x.

3. The proposed algebraic model

In this section, we build a concrete algebraic model to represent access control policies. A policy is an information which we represent as a function. This function can be written as a set of 2-tuples (i,A) where i is an index used to classify an information and A is a set such as $\{(i,A) \mid i \in J \ \wedge \ A \subseteq \mathbb{A}_i\}$, where $\mathbb{A}_i$ is the set of all elements that are classified as i. We use the set Φ to represent the set of all policies that can be defined in an organization, and use the set Φ_p to denote the defined policies in an organization. We specify a policy as the privilege given to a user on an object. Therefore, the set of frames is $I = \{subject, object, privilege\}$. To represent Role Based Access Control RBAC policies, we should add the *Role* frame to the structure. The lattice D constructed from frames is $(\mathscr{P}(I),\cup,\cap)$ and the empty information is defined as $e_J \triangleq \{(i,\emptyset) \mid i \in J\}$. We name the structure $(\Phi_p,(\Phi,D))$ a policy structure.

We present two ways of combining policies. One way relaxes the policies by adding more subjects, objects, and privileges. For example, an organization may specify that Alice is allowed to read from *file*_1, and another organization specifies that Alice is allowed to write into *file*_1. By combining the two policies, we get the policy $\{(subject,\{Alice\}),(object,\{file\_1\}),(privilege,\{read,write\})\}$ that allows Alice to read and write into *file*_1. We define this operator as $\varphi \cdot \psi \triangleq \ \{(i,A) \mid i \in I \ \wedge \ A = \varphi(i) \cup \psi(i)\}$. Usually, we represent it as $\varphi\psi$ (omit the dot).

Another way of combining policies is to restrict the privileges of the two policies by considering the common ones only. For example, by combining the policy $\{(subject, \{Alice\}),(object,\{file\_1,file\_2\}),(privilege,\{read\})\}$ with the policy $\{(subject,\{Alice\}),(object,\{file\_1\}), (privilege,\{read, write\})\}$ we get a more restricted policy that allows Alice to read from *file*_1 $\{(subject,\{Alice\}),(object,\{file\_1\}),(privilege,\{read\})\}$. We define this combination as $\varphi * \psi \triangleq \ \{(i,A) \mid i \in I \ \wedge \ A = \varphi(i) \cap \psi(i)\}$.

We also define a binary operator $\downarrow: \Phi \times D \to \Phi$ to extract a part of an information that belongs to a specific frame as $\varphi^{\downarrow J} \triangleq I_J;\varphi$ where J is a frame and φ is an information such that $J \in D$ and $\varphi \in \Phi$. For example, let $\varphi = \{(subject,\{Alice\}),(object,\{file\_1\}),(privilege,\{read\})\}$ and $J = \{subject\}$. Then $\varphi^{\downarrow J} = \{(subject,\{Alice\})\}$. It is proved in[4] that $((\Phi,\cdot),(D,\curlyvee,\curlywedge),\downarrow,d,e)$ is an information algebra. Therefore, $(\Phi,\cdot)$ has a natural order relation $\leq$ such that $\varphi \leq \psi \Leftrightarrow \varphi \cdot \psi = \varphi$. We use this relation to compare policies based on their flexibility. For example, let φ and ψ be two policies such that $\varphi = \{(subject,\{Alice\}),(object,\{file\_1\}),(privilege,\{read\})\}$, $\psi = \{(subject,\{Alice, Bob\}),(object,\{file\_1\}),(privilege,\{read\})\}$. Here, φ is more restricted than ψ as ψ allows Alice and Bob to read from *file*_1 while φ allows only Alice. Formally, $\varphi \leq \psi$ because $\varphi\psi = \psi$.

An information can be a composite policy. For example, the information $\{(subject,\{Bob\}),(object,\{file\_2\}), (privilege,\{read,write\})\}$ is a composite policy that contains two policies. The first allow Bob to read from *file*_2 and the second allow Bob to write into *file*_2. We call a policy elementary if the set associated with each classification is singleton i.e. elementary(φ) iff $\forall(\psi,\chi \ \mid \ \psi,\chi \in \Phi \ : \ \psi \neq \chi \Rightarrow \psi\chi \neq \varphi)$. From a composite policy, we can extract all its elementary policies as singleton$(\varphi) = \{\psi \mid$ elementary$(\psi) \ \wedge \ \psi \leq \varphi\}$. For example, let the policy $\varphi =$

Download English Version:

https://daneshyari.com/en/article/485416

Download Persian Version:

https://daneshyari.com/article/485416

Daneshyari.com