A Comparison of Logical-Formula and Enumerated Authorization Policy ABAC Models

Prosunjit Biswas, Ravi Sandhu, Ram Krishnan

A Comparison of Logical-formula and Enumerated Authorization Policy ABAC Models

Prosunjit Biswas, Ravi Sandhu, and Ram Krishnan

Institute for Cyber Security
University of Texas at San Antonio
prosun.csedu@gmail.com, {ravi.sandhu, ram.krishnan}@utsa.edu

Abstract. Logical formulas and enumeration are the two major ways for specifying authorization policies in Attribute Based Access Control (ABAC). While considerable research has been done for specifying logical-formula authorization policy ABAC, there has been less attention to enumerated authorization policy ABAC. This paper presents a finite attribute, finite domain ABAC model for enumerated authorization policies and investigates its relationship with logical-formula authorization policy ABAC models in the finite domain. We show that these models are equivalent in their theoretical expressive power. We also show that single and multi-attribute ABAC models are equally expressive.

1 Introduction

Attribute Based Access Control (ABAC) has gained considerable attention from businesses, academia and standard bodies, such as NIST [6], in recent years. ABAC uses attributes on users, objects and possibly other entities (e.g. context or environment) and specifies rules using these attributes to assert who can have which access permissions (e.g. read or write) on which objects. Although ABAC concepts have been around for over two decades there remains a lack of well-accepted ABAC models. Recently there has been a resurgence of interest in ABAC due to continued dissatisfaction with the three traditional models (DAC [14], MAC [12], RBAC [13]), and particularly with the limitations of RBAC.

To demonstrate expressive power and flexibility, several ABAC models including [7, 15, 16, 19] have been proposed in past few years. These models adopt the conventional approach of designing attribute based authorization policies as logical formulas. Logical-formula authorization policies (LAPs) are powerful and convenient to specify even complicated business requirements in a concise way.

An alternate to specify authorization policies is by enumeration, called enumerated authorization policies (EAPs). Examples in this category in-

clude *Policy Machine* (*PM*) [5] and *LaBAC* [2]. These models demonstrate expressiveness by their ability to configure traditional models.

Thus, LAPs and EAPs are two viable approaches to express authorization policies in an ABAC model. While ABAC models with LAPs (denoted *LAP-ABAC*) have received considerable attention, design and development of ABAC with EAPs (denoted *EAP-ABAC*) are relatively neglected. As a result, there is scant literature on development of *EAP-ABAC*. Nonetheless, a comparison between these two approaches is required to further fundamental understanding of ABAC.

This paper presents a finite attribute, finite domain model for *EAP-ABAC* and investigates its relationship with *LAP-ABAC* in the finite domain. We show that *LAP-ABAC* and *EAP-ABAC* are equivalent in theoretical expressive power. We also show that single and multi-attribute models are equally expressive.

Rest of this paper is organized as follows. Section 2 discusses different styles and scopes for ABAC. Section 3 presents multi-attribute *EAP-ABAC* and *LAP-ABAC* models. We show that these models are equivalent in theoretical expressive power in Section 4. Related work is presented in Section 5. Finally, Section 6 concludes the paper.

2 Authorization policy representation

In this section, we discuss two types of authorization policies—logical-formula and enumeration with respect to finite domain ABAC models.

Finite domain ABAC models. Most of the ABAC models (for example, [7, 15, 16, 19]) assume a finite set of user and object attributes and that values of these attributes come from a finite set. This assumption is useful in many practical cases. For example, values of *roles*, *clearance* or *age* are bounded and mostly static. But attribute values can be unbounded as well. For example, if values of an attribute include users or objects in a system (e.g. the attribute *owner* for an object) and these values may grow indefinitely, they are unbounded. This paper focuses on finite-domain ABAC models that have a finite set of attributes with finite ranges for attribute values.

Logical-formula authorization policy. A logical-formula authorization policy is defined as a boolean expression consisting of subexpressions connected with logical operators (for example, $\wedge, \vee, \neg$). These subexpressions compare attribute values with other attribute or constant values. LAPs are usually expressed in propositional logic and support a large set of logical and relational operators. A LAP grants an authoriza-

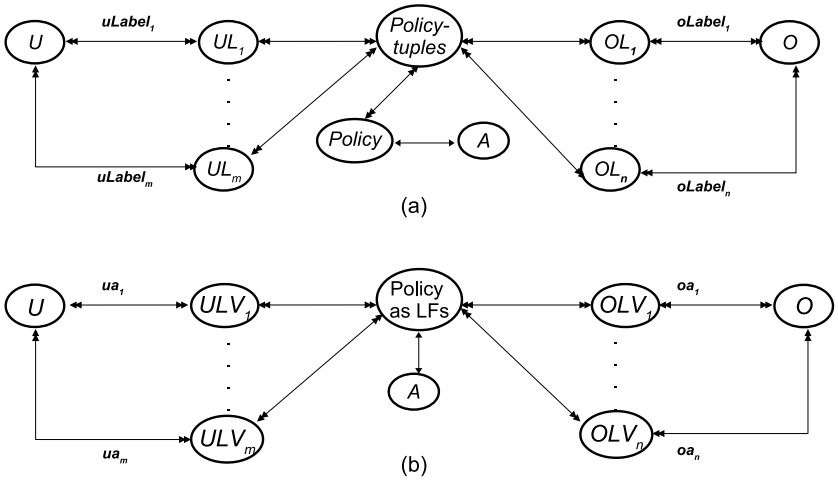

Fig. 1: Components of (a) $EAP\text{-}ABAC_{m,n}$ (b) $LAP\text{-}ABAC_{m,n}$

tion request if the the applicable formula evaluate true for attribute values of the requesting user and requested object. $Auth_{read} \equiv$ *clearance(u)* $\succeq$ *classification(o)* is an example of LAP which allows a user to read an object if the user's clearance dominates classification of the object.

Enumerated authorization policy. An enumerated authorization policy consists of a set of tuples. Each tuple, represented as *(user-attr-values, obj-attr-values)*, grants privileges to a set of users to exercise an action on a set of objects identified by the user and object attribute values mentioned in the tuple. In an EAP, each tuple is distinct and grants privileges independently. User and object attribute values used in the tuple can be atomic or set valued. For example, (*mng, TS*) and ($\{$*mng, dir*$\}$, $\{$*TS,H*$\}$) are atomic and set valued tuples respectively.

3 Finite domain ABAC models

In this section, we define a multi-attribute enumerated authorization policy ABAC model named $EAP\text{-}ABAC_{m,n}$ (shown in Figure 1(a)). To the best of our knowledge, $EAP\text{-}ABAC_{m,n}$ is the first such model. PM[5] also defines a multi-attribute $EAP\text{-}ABAC$ model, but its interpretation of attributes is different than the traditional interpretation of attributes as *(attr. name, value)* pairs. We also define a multi-attribute $LAP\text{-}ABAC$ model named $LAP\text{-}ABAC_{m,n}$ (shown in Figure 1(b)) by abstracting its policy language and potentially accepting any computational logic as policy language.

Table 1: $EAP\text{-}ABAC_{m,n}$ model

I. Sets and relations
- U, O, and A (users, objects and actions respectively)
- $UL_1, UL_2, ...UL_m$ (values for $uLabel_1$, $uLabel_2$, ... , $uLabel_m$)
- $OL_1, OL_2, ...OL_n$ (values for $oLabel_1$, $oLabel_2$, ... , $oLabel_n$)
- $uLabel_i : U \rightarrow 2^{UL_i}$, for $1 \leq i \leq m$;
- $oLabel_i : O \rightarrow 2^{OL_i}$, for $1 \leq i \leq n$
II. Policy components
- *Policy-tuples* $= (2^{UL_1} \times 2^{UL_2} \times ... \times 2^{UL_m}) \times (2^{OL_1} \times 2^{OL_2} \times ... \times 2^{OL_n})$
- $Policy_a \subseteq$ *Policy-tuples* and $Policy = \{Policy_a \mid a \in A\}$
III. Authorization function
- $is\_authorized(u : U, a : A, o : O) = (\exists (ULS_1, ULS_2, ..., ULS_m, OLS_1, OLS_2, ...OLS_n)$ $\in Policy_a)[ULS_i \subseteq uLabel_i(u), \text{for } 1 \leq i \leq m \wedge OLS_i \subseteq oLabel_i(o), \text{for } 1 \leq i \leq n]$

Multi-attribute $EAP\text{-}ABAC$ **(**$EAP\text{-}ABAC_{m,n}$**):** $EAP\text{-}ABAC_{m,n}$ has m user attributes and n object attributes. Components of $EAP\text{-}ABAC_{m,n}$ are shown in Figure 1(a). The unbounded set of users and objects, and finite set of actions are represented by U, O and A respectively. The values denoted by UL_1, UL_2 through UL_m (UL_1 and UL_m are shown in the figure) represent range of m user attribute functions named $uLabel_1, uLabel_2$ through $uLabel_m$ respectively. Similarly, OL_1 OL_2 through OL_n specify values of n object attributes. For simplicity, we do not consider subjects or sessions, distinct from users, here. They do not materially affect the discussion.

The set of policies is represented by *Policy*. We define one policy per action. A policy is defined a set of policy-tuples. A policy-tuple includes subset of values for each user and object attribute.

The formal definition of the model and semantics of the authorization function are given in Table 1. Segment I of the table defines basic sets and relations discussed above. In Segment II, shows notation of policy tuples and defines a policy as subset of tuples. Finally, the authorization function $is\_authorized(s, a, o)$ is presented in Segment III. It allows a user u to perform an action a on an object o if in the policy $Policy_a$ for action a, there exists a tuple that satisfies following conditions—(i) u possesses attribute values used in the tuple, and (ii) o is assigned attribute values mentioned in the tuple.

Multi-attribute $LAP\text{-}ABAC$ **(**$LAP\text{-}ABAC_{m,n}$**):** $LAP\text{-}ABAC_{m,n}$ is specified in Figure 1(b). This model is based on LAPs. Other than authorization policies, this model is similar to $EAP\text{-}ABAC_{m,n}$. It defines a LAP as a boolean function f_a that takes values of m user and n object attributes as arguments. An authorization request for action a is granted if

Table 2: $LAP\text{-}ABAC_{m,n}$ model

I. Sets and relations
- U, O and A (set of users, objects and actions respectively)
- $UAV_1, UAV_2, ..., UAV_m$ (range of user attribute functions)
- $OAV_1, OAV_2, ..., OAV_n$ (range of object attribute functions)
- $UA = \{ua_1, ua_2, ..., ua_m\}$ (set of user attributes); $ua_i : U \rightarrow 2^{UAV_i}$, for $1 \leq i \leq m$
- $OA = \{oa_1, oa_2, ..., oa_n\}$ (set of object attributes); $oa_i : O \rightarrow 2^{OAV_i}$, for $1 \leq i \leq n$
II. Policy components
- $f_a : (2^{UAV_1}, ..., 2^{UAV_m}, 2^{OAV_1}, ..., 2^{OAV_n}) \rightarrow \{true, false\}$ (policy for $a \in A$).
- $LFs = \{f_a \mid a \in A\}$ (set of all policies)
III. Authorization function
- is_authorized(u:U,a:A,o:O) = $\exists f_a \in LFs[f_a(ua_1(u), ua_2(u), ..., ua_m(u), oa_1(o), oa_2(o), ...oa_n(o)) = true]$

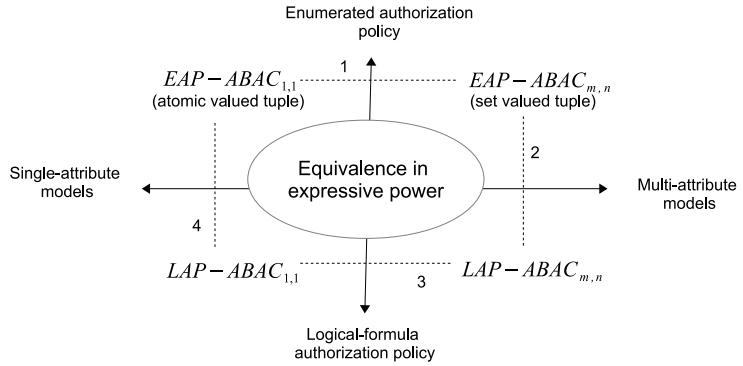

Fig. 2: Equivalence of EAP and LAP ABAC models

$f_a()$ is evaluated true for attribute values of requesting user and requested object. The formal definition is given in Table 2, similar to Table 1.

4 Theoretical expressive power of EAP and LAP models

This section establishes equivalence between different *EAP-ABAC* and *LAP-ABAC* models with respect to their theoretical expressive power. We consider single and multi-attribute *EAP-ABAC* and *LAP-ABAC* models. The relationship among the models we consider is schematically presented in Figure 2. Single attribute and multi-attribute models are presented on left and right side of the Y-axis respectively. Enumerated and logical-formula policy models are presented above and below the X-axis respectively. These models all have set valued attributes. Policy tuples are represented differently in $EAP\text{-}ABAC_{1,1}$ and $EAP\text{-}ABAC_{m,n}$ models.

Table 3: Mappings

Equivalence of $EAP\text{-}ABAC_{m,n}$ ***and*** $EAP\text{-}ABAC_{1,1}$

I. From $EAP\text{-}ABAC_{m,n}$ *to* $EAP\text{-}ABAC_{1,1}$

- $U = U, O = O, A = A$
- $UL = 2^{UL_1} \times 2^{UL_2} \times ... \times 2^{UL_m}$; $OL = 2^{OL_1} \times 2^{OL_2} \times ... \times 2^{OL_m}$
- $uLabel(u) = 2^{uLabel_1(u)} \times 2^{uLabel_2(u)} \times ... \times 2^{uLabel_m(u)}$
- $oLabel(u) = 2^{oLabel_1(o)} \times 2^{oLabel_2(o)} \times ... \times 2^{oLabel_n(o)}$
- $Policy_{a_{1,1}} = \{((ULS_1, ULS_2, ..., ULS_m), (OLS_1, OLS_2, ..., OLS_n))|$
  $(\exists (ULS_1, ULS_2, ..., ULS_m, OLS_1, OLS_2, ..., OLS_n) \in Policy_a)\ [Policy_a \in Policy_{m,n}]\}$

II. From $EAP\text{-}ABAC_{1,1}$ *to* $EAP\text{-}ABAC_{m,n}$

- $EAP\text{-}ABAC_{1,1}$ is a special case of $EAP\text{-}ABAC_{m,n}$.

Equivalence of $EAP\text{-}ABAC_{m,n}$ ***and*** $LAP\text{-}ABAC_{m,n}$

III. From $EAP\text{-}ABAC_{m,n}$ *to* $LAP\text{-}ABAC_{m,n}$

- $UAV_i = UL_i$, for $1 \leq i \leq m$; $OAV_i = OL_i$, for $1 \leq i \leq n$
- $ua_i(u) = uLabel_i(u)$; $oa_i(o) = oLabel_i(o)$
- $f_a =$
$$\bigvee_{(ULS_1, ULS_2, ..ULS_m, OLS_1, OLS_2, .., OLS_n) \in Policy_a} (\bigwedge_{1 \leq i \leq m} ULS_i \subseteq ua_i(u)) \wedge (\bigwedge_{1 \leq i \leq n} OLS_i \subseteq oa_i(u)$$

IV. From $LAP\text{-}ABAC_{m,n}$ *to* $EAP\text{-}ABAC_{m,n}$

- $UL_i = UAV_i$, for $1 \leq i \leq m$; $OL_i = OAV_i$, for $1 \leq i \leq n$
- $uLabel_i(u) = ua_i(u)$, for $1 \leq i \leq m$; $oLabel_i(o) = oa_i(o)$, for $1 \leq i \leq n$
- $Policy_a = \{(ULS_1, ULS_2, ..., ULS_m, OLS_1, OLS_2, ..., OLS_n)|$
  $f_a(ULS_1, ULS_2, ..., ULS_m, OLS_1, OLS_2, ..., OLS_n) = true\}$

Equivalence of $LAP\text{-}ABAC_{m,n}$ ***and*** $LAP\text{-}ABAC_{1,1}$

V. From $LAP\text{-}ABAC_{m,n}$ *to* $LAP\text{-}ABAC_{1,1}$

- $U = U_{m,n}; O = O_{m,n}; A = A_{m,n}; UAV = UAV_1 \times UAV_2 \times ... \times UAV_m$
- $OAV = OAV_1 \times OAV_2 \times ... \times OAV_m; ua(u) = ua_1(u) \times ua_2(u) \times ... \times ua_m(u)$
- $oa(u) = oa_1(u) \times ... \times oa_m(u)$
- $f_a = \bigvee_{f_{a_{m,n}}(ULS_1, ULS_2, ...ULS_m, OLS_1, OLS_2, ..., OLS_n) = true} (ULS_1(u) \times ... \times ULS_m(u) \subseteq ua(u)) \wedge$
  $(OLS_1(o) \times ... \times OLS_n(o) \subseteq oa(o))$, for $ULS_i \subseteq UAV_i$ and $OLS_i \subseteq OAV_i$

VI. From $LAP\text{-}ABAC_{1,1}$ *to* $LAP\text{-}ABAC_{m,n}$

- $LAP\text{-}ABAC_{1,1}$ is a special case of $LAP\text{-}ABAC_{m,n}$.

Equivalence of $EAP\text{-}ABAC_{1,1}$ ***and*** $LAP\text{-}ABAC_{1,1}$

VII & VIII. From $EAP\text{-}ABAC_{1,1}$ *to* $LAP\text{-}ABAC_{1,1}$ *and vice versa*

- Special case of equivalence of $EAP\text{-}ABAC_{m,n}$ and $EAP\text{-}ABAC_{1,1}$.

The former uses atomic valued tuples (e.g. $(manager, TS)$) and the later uses set valued tuples (e.g. $(\{manager\}\,\{TS\})$).

Four different equivalences are discussed here labeled one to four in Figure 2. They are equivalence of (i) single and multi-attribute EAP models, (ii) multi-attribute EAP and LAP models, (iii) single and multi-attribute LAP models, and (iv) single attribute LAP and EAP models.

The equivalence of single and multi-attribute EAP models are demonstrated in Segment I and II in Table 3. In Segment I, we show that multiple attributes can be represented as a single attribute comprising of cross product of values of multiple attributes. Segment II is trivial as $EAP\text{-}ABAC_{1,1}$ is a special case of $EAP\text{-}ABAC_{m,n}$. Segment III shows how to construct a LAP formula using m user and n object attributes from a enumerated policy of same set of attributes. Segment IV shows the converse. Similar to Segment I, Segment V shows how a logical formula of multiple user and object attributes can be represented as a logical formula of single user and object attributes. Segment VI is trivial as $LAP\text{-}ABAC_{1,1}$ is a special case of $LAP\text{-}ABAC_{m,n}$. The equivalence of single attribute EAP and LAP models presented in Segment VII and VIII is a special case of the equivalence of multi-attribute EAP and LAP models presented in Segment III and IV.

5 Related work

Several ABAC models have been proposed in the literature. Most of them are based on LAPs. For example, ABAC_α [7] is among the first few models to formally define a $LAP\text{-}ABAC$. HGABAC [15] is a more general purpose $LAP\text{-}ABAC$ model. Other works include [8, 11, 16, 18, 19].

Damiani et al [4] describe an informal framework for attribute based access control in open environments. Bonatti et al [3] present a uniform structure to logically formulate and reason about both service access and information disclosure constraints according to related entity attributes. NIST ABAC guide [6] is significant in defining concepts, required components, considerations and architecture for designing an enterprise ABAC system. Other notable works include XACML [9], UCON [10] and Armando et al. [1].

6 Conclusion

We have presented a finite attribute, finite domain ABAC model using enumerated authorization policies. We show that enumerated authorization policy and logical-formula authorization policy ABAC models are

equivalent in their theoretical expressive power. We believe, analysis of these two models beyond expressive power is required to better understand these models and ABAC in general.

7 Acknowledgement

This research is partially supported by NSF Grants CNS-1111925 and CNS-1423481.

References

1. A. Armando et al. SMT-based enforcement and analysis of NATO content-based protection and release policies. In *ABAC '16*, pages 35–46. ACM, 2016.
2. P. Biswas, R. Sandhu, and R. Krishnan. Label-based access control: An ABAC model with enumerated authorization policy. In *ABAC '16*. ACM, 2016.
3. P. Bonatti and P. Samarati. Regulating service access and information release on the web. In *Proceedings of CCS*, pages 134–143. ACM, 2000.
4. E. Damiani, S. D. C. di Vimercati, and P. Samarati. New paradigms for access control in open environments. In *Sig. Proc. and Info. Tech., 2005*.
5. D. Ferraiolo et al. The Policy Machine: A novel architecture and framework for access control policy specification and enforcement. *JSA*, 57(4):412–424, 2011.
6. V. C. Hu et al. Guide to attribute based access control (ABAC) definition and considerations. *NIST Special Publication*, 800:162, 2014.
7. X. Jin, R. Krishnan, and R. S. Sandhu. A unified attribute-based access control model covering DAC, MAC and RBAC. *DBSec*, 12:41–55, 2012.
8. B. Lang et al. A flexible attribute based access control method for grid computing. *Journal of Grid Computing*, 7(2):169–180, 2009.
9. T. Moses et al. Extensible access control markup language (XACML) version 2.0. *Oasis Standard*, 2005.
10. J. Park and R. Sandhu. The UCON ABC usage control model. *TISSEC*, 2004.
11. T. Priebe, W. Dobmeier, and N. Kamprath. Supporting attribute-based access control with ontologies. In *ARES 2006*, pages 8–pp. IEEE, 2006.
12. R. S. Sandhu. Lattice-based access control models. *Computer*, 26(11):9–19, 1993.
13. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *Computer*, (2):38–47, 1996.
14. R. S. Sandhu and P. Samarati. Access control: principle and practice. *Communications Magazine, IEEE*, 32(9):40–48, 1994.
15. D. Servos and S. L. Osborn. HGABAC: Towards a formal model of hierarchical attribute-based access control. In *FPS*, pages 187–204. Springer, 2014.
16. H.-b. Shen and F. Hong. An attribute-based access control model for web services. In *PDCAT'06.*, pages 74–79. IEEE, 2006.
17. M. V. Tripunitara and N. Li. A theory for comparing the expressive power of access control models. *Journal of Computer Security*, 15(2):231–272, 2007.
18. L. Wang, D. Wijesekera, and S. Jajodia. A logic-based framework for attribute based access control. In *Proceedings. FMSE '04*, pages 45–55. ACM, 2004.
19. E. Yuan and J. Tong. Attributed based access control (ABAC) for web services. In *Proceedings. 2005 IEEE International Conference on Web Service*. IEEE, 2005.