The Authorization Policy Existence Problem

Pierre Bergé
LRI, Université Paris-Saclay
Bât 650, Rue Noetzlin
91190 Gif-sur-Yvette
France
pierre.berge@supelec.fr

Jason Crampton
Royal Holloway
University of London
Egham, TW20 9QY
United Kingdom
jason.crampton@rhul.ac.uk

Gregory Gutin
Royal Holloway
University of London
Egham, TW20 9QY
United Kingdom
gutin@cs.rhul.ac.uk

Rémi Watrigant
INRIA Sophia-Antipolis
2004 Route des Lucioles
06902 Sophia-Antipolis
France
remi.watrigant@inria.fr

ABSTRACT

Constraints such as separation-of-duty are widely used to specify requirements that supplement basic authorization policies. However, the existence of constraints (and authorization policies) may mean that a user is unable to fulfill her/his organizational duties because access to resources is denied. In short, there is a tension between the need to protect resources (using policies and constraints) and the availability of resources. Recent work on workflow satisfiability and resiliency in access control asks whether this tension compromises the ability of an organization to achieve its objectives. In this paper, we develop a new method of specifying constraints which subsumes much related work and allows a wider range of constraints to be specified. The use of such constraints leads naturally to a range of questions related to "policy existence", where a positive answer means that an organization's objectives can be realized. We provide an overview of our results establishing that some policy existence questions, notably for those instances that are restricted to user-independent constraints, are fixed-parameter tractable.

CCS Concepts

•**Security and privacy** → **Access control;** *Security requirements;* •**Theory of computation** → **Fixed parameter tractability;**

Keywords

access control; resiliency; satisfiability; computational complexity; fixed-parameter tractability

CODASPY'17, March 22-24, 2017, Scottsdale, AZ, USA
ACM 978-1-4503-4523-1/17/03.
http://dx.doi.org/10.1145/3029806.3029844

1. INTRODUCTION

Access control is a fundamental aspect of the security of any multi-user computing system, and is typically based on the specification and enforcement of an authorization policy. Such a policy identifies which interactions between users and resources are to be allowed by the system.

Over the last twenty years, access control requirements have become increasingly complex, leading to increasingly sophisticated authorization policies, often expressed in terms of constraints. A separation-of-duty constraint (also known as the "two-man rule" or "four-eyes policy") may, for example, require that no single user is authorized for some particularly sensitive group of resources. Such a constraint is typically used to prevent misuse of the system by a single user.

The use of authorization policies and constraints, by design, limits which users may access resources. Nevertheless, the ability to perform one's duties requires access to particular resources, and overly prescriptive policies and constraints may mean that some resources are inaccessible. In short, "tension" may exist between authorization policies and operational demands: too lax a policy may suit organizational demands but lead to security violations; whereas too restrictive a policy may compromise an organization's ability to meet its business objectives.

Recent work on workflow satisfiability and access control resiliency has recognized the importance of being able to identify whether or not security policies prevent an organization from achieving its objectives [1, 5, 6, 9, 11]. In this paper, we seek to generalize existing work in this area. Specifically, we introduce the Authorization Policy Existence Problem (APEP). Informally, APEP seeks to find an authorization policy, subject to restrictions on individual authorizations (defined by a "base" authorization relation) and restrictions on collective authorizations (defined by a set of authorization constraints).

The framework within which APEP is defined admits a greater variety of constraints than is usually considered in either the standard access control literature [3, 7, 8, 10] or in workflow satisfiability [1, 4, 11]. In this paper we characterize the constraints of interest and extend the definition of user-independent constraints [4] to this framework. We

conclude the paper by stating some results for the complexity of APEP when problem instances are characterized by the types of constraints that may be included.

2. THE AUTHORIZATION POLICY EXISTENCE PROBLEM

In this paper, we extend existing work on workflow satisfiability, constraints and resiliency, by defining a simple yet very expressive authorization framework. Roughly speaking, we specify a problem dealing with the existence of an appropriate authorization relation.

Given a set of users U and a set of resources R to which access should be restricted, we may define an *authorization relation* $A \subseteq U \times R$, where $(u, r) \in A$ if and only if u is authorized to access r. Given a resource r, we will write $A(r)$ to denote the set of users that are authorized to access resource r. More formally, $A(r) = \{u \in U : (u, r) \in A\}$. Similarly, for $u \in U$, we will write $A(u)$ to denote the set of resources that u is authorized to access, that is $A(u) = \{r \in R : (u, r) \in A\}$. We extend this notation to subsets of R and U in the natural way: for $R' \subseteq R$ and $U' \subseteq U$,

$$A(R') \stackrel{\text{def}}{=} \bigcup_{r \in R'} A(r) \quad \text{and} \quad A(U') \stackrel{\text{def}}{=} \bigcup_{u \in U'} A(u).$$

The Authorization Policy Existence Problem is defined with reference to (i) a *base authorization relation* $A_{\mathsf{Bse}} \subseteq U \times R$ such that $A_{\mathsf{Bse}}(r) \neq \emptyset$ for each $r \in R$, and (ii) a set of *authorization constraints* C. Informally, A_{Bse} specifies restrictions on all valid authorization relations, while C specifies additional restrictions that any valid authorization relation must satisfy. We discuss constraints in more detail in Section 3.

Formally, given a base authorization relation A_{Bse} and a set of constraints C, we say an authorization relation $A \subseteq U \times R$ is

- *authorized* with respect to A_{Bse} if $A \subseteq A_{\mathsf{Bse}}$;
- *complete* if $A(r) \neq \emptyset$ for every $r \in R$;
- *eligible* with respect to C if A satisfies c for all $c \in C$;
- *valid* with respect to A_{Bse} and C if A is authorized, complete and eligible.

Given A_{Bse} and C, the decision Authorization Policy Existence Problem (APEP) asks whether there exists a valid authorization relation.

We assume that determining whether an authorization relation satisfies a constraint takes polynomial time. (This is a reasonable assumption for all constraints of relevance to access control.) Let n denote $|U|$, k denote $|R|$ and m denote $|C|$. Then a brute force approach to solving APEP (by simply examining every possible authorization relation) takes time $O^*(2^{nk})$. (The O^* notation ignores multiplicative polynomial terms.)

3. CONSTRAINTS

We first introduce constraints that generalize separation-of-duty and binding-of-duty constraints. They are defined in terms of pairs of resources r and r', and their satisfaction is defined by a relationship that must hold on $A(r)$ and $A(r')$.

1. $(r, r', \leftrightarrow, \exists)$ is satisfied if there exists $u \in U$ such that $u \in A(r)$ and $u \in A(r')$; that is, $A(r) \cap A(r') \neq \emptyset$.
2. $(r, r', \updownarrow, \exists)$ is satisfied if there exists $u \in U$ such that either (i) $u \in A(r)$ and $u \notin A(r')$ or (ii) $u \notin A(r)$ and $u \in A(r)$; that is, $A(r) \neq A(r')$.
3. $(r, r', \leftrightarrow, \forall)$ is satisfied if for all $u \in A(r) \cup A(r')$, $u \in A(r)$ if and only if $u \in A(r')$; that is, $A(r) = A(r')$.
4. $(r, r', \updownarrow, \forall)$ is satisfied if for all $u \in A(r) \cup A(r')$, either (i) $u \in A(r)$ and $u \notin A(r')$ or (ii) $u \notin A(r)$ and $u \in A(r')$; that is, $A(r) \cap A(r') = \emptyset$.

Constraints of the form $(r, r', \updownarrow, Q)$ correspond closely to the idea of separation-of-duty. Indeed, the satisfaction criterion for $(r, r', \updownarrow, \forall)$ is identical to that for a simple static separation-of-duty constraint. Similarly, constraints of the form $(r, r', \leftrightarrow, Q)$ correspond to the idea of binding-of-duty.

A constraint of the form $(r, r', \circ, \forall)$ is said to be *universal*, while a constraint of the form $(r, r', \circ, \exists)$ is said to be *existential*. Informally speaking, universal constraints are "stronger" than existential constraints: for any complete relation, the satisfaction of $(r, r', \sim, \forall)$ implies the satisfaction of $(r, r', \sim, \exists)$, but the converse does not hold.

3.1 Cardinality constraints

We may also define *cardinality constraints*, which come in two flavors. In the following, $\lhd$ is one of $=$, $<$, $>$, $\leqslant$ or $\geqslant$ and t is an integer greater than 0.

- A *global* (cardinality) constraint has the form $(\lhd, t)$. The constraint $(\lhd, t)$ is satisfied by relation A if for all $r \in R$, $|A(r)| \lhd t$.
- A *local* (cardinality) constraint has the form $(R', \lhd, t)$, where $R' \subseteq R$. The constraint $(R', \lhd, t)$ is satisfied by relation A if $|A(R')| \lhd t$.

Then, for example, the global constraint $(=, 1)$ requires a valid relation A to be a function (since the number of users assigned to each resource is precisely 1), while the local constraint $(\{r\}, \leqslant, t)$ is a cardinality constraint in the RBAC96 sense [10] (if resource r is interpreted as a role). Finally, the t-out-of-m static separation-of-duty constraint $\mathsf{ssod}(\{r_1, \ldots, r_m\}, t)$, introduced by Li *et al.* [8], may be represented by the cardinality constraint $(\{r_1, \ldots, r_m\}, \geqslant, t)$.

3.2 User-independent constraints

User-independent (UI) constraints are important in the context of the Workflow Satisfiability Problem (WSP) [4]. First, the class of UI constraints includes a very wide range of constraints, and almost all constraints that are of relevance to access control. Second, WSP is fixed-parameter tractable (FPT) if we restrict attention to UI constraints [4]. (WSP is not FPT if we allow arbitrarily complex constraints [11].) Informally, a constraint is UI in the context of workflow satisfiability if its satisfaction only depends on the relationships that exist between users assigned to steps in a workflow (and not on the specific identities of users) [4].

Let A be an authorization relation and $\sigma : U \to U$ a permutation of the user set (that is, σ is a bijection). Then, given an authorization relation $A \subseteq U \times R$, we write $\sigma(A) \subseteq U \times R$ to denote the relation $\{(\sigma(u), r) : (u, r) \in A\}$. A constraint c is *user-independent* if for every authorization relation A that satisfies c and every permutation $\sigma : U \to U$, $\sigma(A)$ satisfies c.

Elementary arguments may be used to show that constraints of the form $(r, r', \leftrightarrow, \exists)$ $(r, r', \leftrightarrow, \forall)$, $(r, r', \updownarrow, \exists)$ and $(r, r', \updownarrow, \forall)$ are UI. Equally, it is clear that global and local constraints, whose satisfaction is defined in terms of the cardinality of sets of the form $A(r)$, are UI, since a permutation will preserve the cardinality of such sets. In other words, all constraints we consider in this paper are UI.

3.3 Bounded UI constraints

We now define an important class of UI constraints that is useful for establishing positive results for variants of APEP. Given a base relation A_{Bse} and a constraint c, let A be valid with respect to A_{Bse} and c. We say A *requires* v if $\{(u,r) \in A : u \neq v\}$ is not valid. (Since A is valid, this means that $\{(u,r) \in A : u \neq v\}$ is either incomplete or does not satisfy c.) Then we define

$$\mathsf{core}(A : A_{\mathsf{Bse}}, c) \stackrel{\text{def}}{=} \{u \in U : A \text{ requires } u\}$$

to be the *core of A with respect to A_{Bse} and c.*

PROPOSITION 1. *Let $\mathcal{I} = (A_{\mathsf{Bse}}, C)$ be a satisfiable instance of* APEP *with a UI constraint $c \in C$. If A is a valid solution with respect to A_{Bse} and c then*

$$|\mathsf{core}(A : U \times R, c)| \geqslant |\mathsf{core}(A : A_{\mathsf{Bse}}, c)|$$

DEFINITION 2. *We say a UI constraint c is $f(k,n)$-*bounded *if $|\mathsf{core}(A : U \times R, c)| \leqslant f(k,n)$ for all A valid with respect to $U \times R$ and c.*

The definition of $f(k,n)$-bounded constraints and Proposition 1 impose an upper bound on the number of users we need to consider when constructing candidate solutions to an instance (A_{Bse}, C) of APEP.

Table 1 shows $f(k,n)$ for different types of constraints. Note that in all cases, $f(k,n)$ is independent of n. This is important as we are able to prove that APEP is FPT when all constraints are $f(k)$-bounded for some function f. The proofs of the above results, and those in Section 4, can be found in the extended version of this paper [2].

Constraint Type	Largest Core
$(r, r', \updownarrow, \forall)$, $(r, r', \updownarrow, \exists)$	k
$(r, r', \leftrightarrow, \forall)$, $(r, r', \leftrightarrow, \exists)$	$k-1$
$(R', \leqslant, t)$	k
$(R', =, t)$, $(R', \geqslant, t)$	$2\max\{k, t\}$

Table 1: Upper bounds on the size of the core

4. COMPLEXITY RESULTS

It is straightforward to show that an instance of APEP containing only constraints of the form $(r, r', \updownarrow, Q)$, where $Q \in \{\exists, \forall\}$, and the constraint $(=, 1)$ is equivalent (in an appropriate complexity-theoretic sense) to an instance of WSP involving only separation-of-duty constraints. In short, APEP is at least as hard as WSP. An instance of WSP is parameterized by the number of workflow steps k, the number of users n, and the number of constraints c. It is known that WSP is fixed-parameter tractable (FPT) for UI constraints [4], where k is the small parameter. That is, informally, there exists an algorithm to solve the problem whose run-time is only exponential in k, which is generally an order of magnitude smaller than n in practice.

We have established the APEP is also FPT if we restrict our attention to particular types of constraints and the number of resources k is the small parameter (relative to the number of users n). Figure 1 summarizes our results for APEP, determined by constraint type. We write $(\circ, Q)$, where $Q \in \{\exists, \forall\}$ and $\circ \in \{\leftrightarrow, \updownarrow\}$, to denote that only constraints of the form $(r, r', \circ, Q)$ are permitted.

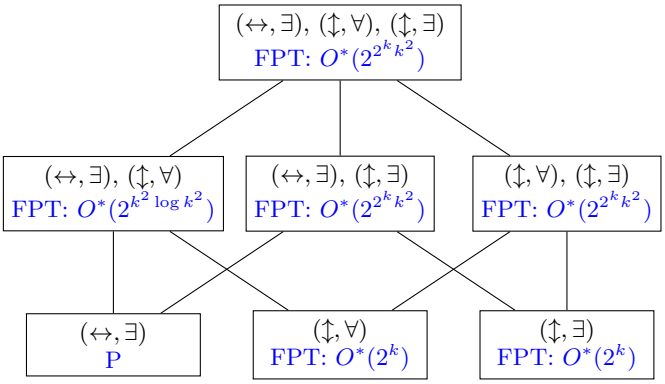

Figure 1: The complexity of variants of APEP

5. REFERENCES

[1] BASIN, D. A., BURRI, S. J., AND KARJOTH, G. Obstruction-free authorization enforcement: Aligning security and business objectives. *Journal of Computer Security 22*, 5 (2014), 661–698.

[2] BERGÉ, P., CRAMPTON, J., GUTIN, G., AND WATRIGANT, R. The authorization policy existence problem. *CoRR abs/1612.06191* (2016).

[3] BREWER, D. F. C., AND NASH, M. J. The Chinese wall security policy. In *Proceedings of the 1989 IEEE Symposium on Security and Privacy* (1989), pp. 206–214.

[4] COHEN, D., CRAMPTON, J., GAGARIN, A., GUTIN, G., AND JONES, M. Iterative plan construction for the workflow satisfiability problem. *J. Artif. Intell. Res. (JAIR) 51* (2014), 555–577.

[5] CRAMPTON, J., GUTIN, G., AND WATRIGANT, R. Resiliency policies in access control revisited. In *Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies* (2016), ACM, pp. 101–111.

[6] CRAMPTON, J., GUTIN, G., AND YEO, A. On the parameterized complexity and kernelization of the workflow satisfiability problem. *ACM Trans. Inf. Syst. Secur. 16*, 1 (2013), 4.

[7] GLIGOR, V. D., GAVRILA, S. I., AND FERRAIOLO, D. F. On the formal definition of separation-of-duty policies and their composition. In *Security and Privacy - 1998 IEEE Symposium on Security and Privacy, Proceedings* (1998), IEEE Computer Society, pp. 172–183.

[8] LI, N., TRIPUNITARA, M. V., AND BIZRI, Z. On mutually exclusive roles and separation-of-duty. *ACM Trans. Inf. Syst. Secur. 10*, 2 (2007).

[9] LI, N., WANG, Q., AND TRIPUNITARA, M. V. Resiliency policies in access control. *ACM Trans. Inf. Syst. Secur. 12*, 4 (2009).

[10] SANDHU, R. S., COYNE, E. J., FEINSTEIN, H. L., AND YOUMAN, C. E. Role-based access control models. *IEEE Computer 29*, 2 (1996), 38–47.

[11] WANG, Q., AND LI, N. Satisfiability and resiliency in workflow authorization systems. *ACM Trans. Inf. Syst. Secur. 13*, 4 (2010), 40.