Connected Protection Structures Via Fuzzy Relations

Pratit Santiprabhob (pratit@cs.fsu.edu)
L.J. Kohout (kohout@cs.fsu.edu)
Department of Computer Science B-173
The Florida State University
Tallahassee, FL 32306-4019

April 9, 1991

Abstract

A concept of connected protection structures is proposed. This new concept is an extension of the original concept of dynamic protection structure to govern inter-system activities in addition to activities within a system. The proposed structures admit both crisp and fuzzy relations. However, due to the superior competence in representing the real-world situations and the fact that crisp relations are just special cases of fuzzy ones, fuzzy relations are to be used in the structures. After the proposed concept has been laid out, the analysis of the connected protection structures is discussed.

1 Introduction

The concept of *Dynamic Protection Structure*, which has been developed in [7,8] in order to solve some problems of dynamics of protection systems posed in [4], is extended in this paper. The protection structure is constructed based on actions of participants in a system. The papers of Kohout and Gaines [7,8] incorporated the original *Protection Matrix* model of Graham and Denning [4] retaining *capabilities*, but substantially extending it by introducing *passes* and *permits*. The protection structure has later been fuzzified in [1,5,6]. The structure is designed to systematically detect and identify *harmful* actions that are possible to occur in the system.

As an extension of the original concept, the concept of **Connected Protection Structures** (CPSs) is proposed. CPSs are to deal with a situation where there are multiple

independent systems, each with its own protection structure already defined, that are *connected together* in order to achieve some *common objectives*. Such connected systems exist in large numbers in the real world as an example one can take a pool of airlines, a coalition of armed forces, a network of computer systems, etc.

The connections of these systems naturally change from time to time, while the systems continue to exist as independent but interacting entities. We certainly do not want to design a global protection structure for a group of connected systems, since such global structure should be quite difficult to analyze because of the magnitude of actions and participants, as well as to modify when changes in policies occur. We would instead want to retain the original protection structures of the participating systems, then create an extra structure for each system to govern inter-system actions. Most of the time only these extra structures are to be changed when there occur changes in the connections. The connected protection structures which consist of both the original protection structures and the extra structures will be described in more details in Section 2.

It is to be emphasized that the CPSs discussed in this paper are conceptual structures abstracted from the systems under consideration based on actions of participants in the systems. The structures are totally separated from the underlying hardware and/or software that implements mechanisms to enforce protection criteria embedded in the structures.

2 Connected Protection Structures

For each connected system, we can define a crisp set $P = \{p_1, p_2, p_3, ...\}$ of *Participants* which consists of possible overlapping crisp set $S = \{s_1, s_2, s_3, ...\}$ of *Subjects* and crisp set $O = \{o_1, o_2, o_3, ...\}$ of *Objects*, as well as a crisp set $A = \{a_1, a_2, a_3, ...\}$ of *Actions*. An action is performed by a subject on an object (which could well be another subject or the original subject itself).

A connected protection structure that belongs to one of the connected systems will have two substructures: an *Internal* structure that deals with actions performed among participants within the system, and an *External* structure that governs actions involving participants of another systems. In each of the two structures there will further be another two substructures: a *Capability Structure* (CS) which says what *can* be done in the system, and a *Forbidding Structure* (FS) which says what *should not* be done in the system. Both capability and forbidding structures are in turn composed of three corresponding relations. The structures discussed here admit both crisp and fuzzy relations. Since fuzzy relations are superior to the crisp counterparts in representing real-word situations and in fact the crisp relations are just special cases of the fuzzy ones, we will discuss in term of fuzzy relations in the rest of this paper.

The use of fuzzy relations here is justified by the fact that the relations in both capability and forbidding structure represent protection policies that is originally stated in natural language. Those policies cannot sensibly account precisely for every action of every participant of any non-trivial system, i.e. they cannot utterly say "yes, can perform" or "no, cannot perform" for every action. Hence for us to elicit what is stated in the policies, and represent this in the relations of CS and FS in qualitative terms of *can* and *should not* on relevant actions of participants, the fuzzy values in the relations come to our rescuse. The three fuzzy relations for capability structure are as follows.

Capability relation: $CAP \subseteq^1 A \times S \times O$ is
"For an action _, a subject _ *can* perform on an object _, to degree _"

Pass relation: $PASS \subseteq A \times S \times S \times O$ is
"For an action _, a subject _ *can* pass to a subject _ a capability to perform on an object _, to degree _"

Permit relation: $PERM \subseteq A \times S \times S \times O$ is
"For an action _, a subject _ *can* transfer to a subject _ the privilege of passing a capability to perform on an object _, to degree _"

The three levels represented by the above three relations reflect our desire to treat a capability as a special kind of object, so that the distribution of capabilities will be systematically governed by the relation $PASS$ and $PERM$. We, therefore, have $PERM$ govern the modifications on $PASS$, where $PASS$ in turn governs the modifications on CAP of the same substructures.

In the forbidding structure, there are the following three relations corresponding to the three relations of the capability structure.

Forbidden action relation: $FACT \subseteq A \times S \times O$

Forbidden pass relation: $FPASS \subseteq A \times S \times S \times O$

Forbidden permit relation: $FPERM \subseteq A \times S \times S \times O$

The meaning of these relations is as that of the corresponding relations in the capability structure with the substitution of *should not* for *can*. It is to be noted at this point that the complement of fuzzy degree in the relations of CS represents the degree of *cannot* while that of FS should be interpreted as *don't care*.

The construction of the external part of each CPS, so called external-CPS, will be based on the protection policies described in the *Mutual Agreements* among the connected systems.

[1] is to be interpreted as a fuzzy subset of.

On the other hand, the internal-CPS would be the original local protection structure that reflects the internal protection policies with a few likely modifications due to the mutual agreements.

In our design of CPSs, the analysis tasks for each connected system are confined to its internal- and external-CPSs which can be performed relatively independent of other systems that do not directly contribute to the external-CPSs. Moreover, whenever changes occur, only a group of affected systems needs to be reanalyzed. In other words, in contrast to the global approach, the CPSs are distributed in nature and the analysis, as well as modifications, can be localized to just the relevant connected systems.

3 Potential Violations

Once the relations are in place, we can modify the CAP and $PASS$ relations according to what are allowed by $PASS$ and $PERM$ relations respectively. These modifications could be based on the anticipated behavior of the participants, or in absence of any anticipations, the worst case scenario. The anticipations concerning actions governed by external-CPSs could be implicit in the mutual agreements. The specific operation to be used in modifying the relations are presented in Section 4.

The *Potential Violations* (PVs) can then be computed for each of the internal- and external-CPSs. The first type of potential violations, the potential *direct*-violations (PDVs), are defined in term of the following three relations.

Potential violating action relation: $PVACT = CAP \sqcap FACT$

Potential violating pass relation: $PVPASS = PASS \sqcap FPASS$

Potential violating permit relation: $PVPERM = PERM \sqcap FPERM$

The second type of violations namely the potential *indirect*-violations (PIVs) is harder to compute. We will first need to obtain the following relation.

Potential indirect action relation: $PIACT \subseteq A \times S \times O$ is
"For an action _, a subject _ *can indirectly* perform on an object _, to degree _"

Then we have a relation representing the potential indirect-violations as follows.

Potential violating indirect action relation: $PVIACT = PIACT \sqcap FACT$

An indirect action is caused by a sequence of combined actions allowed by CAP. We need

to analyze actions that are allowed by both internal- and external-CAPs to identify such sequences. Note that internal-, as well as external-PIVs, can be caused by a sequence that contains both actions allowed by internal-CAP and those allowed by external-CAP.

The potential violations represent holes in the protection policies. The holes that cause these PVs will need to be plugged, and the CPSs need to be adjusted accordingly. Only when no major potential violations exist, the connected systems are considered safe and stable.

4 Analysis of CPSs

The very first thing to do before we can perform any analysis on the CPSs is to construct the CPSs themselves. Translation of the protection policies (implied in the mutual agreement) into the relations of CPSs involves natural language interpretation. Various techniques can be used depending on nature of the application ranging from having a human read the agreements and subjectively assign fuzzy values for the relations to having sophisticated computer systems do the job. A simple technique based on *Checklist Paradiagm* [2,3] is being investigated by the authors.

Assuming that we have already obtained all the initial-state fuzzy relations of the CPSs, we then want to manipulate the relations of the capability structure so that the effects of the dynamic distribution of capablilies can be analyzed. In the rest of this Section we define semantics of modifications rules for CAP and $PASS$, and then a new operator to perform the modifications on the relations is introduced.

In the process of modifications, we want to *copy* capabilities and passes possessed by one subject to another subject based on what is allowed by $PASS$ and $PERM$ respectively. Let $cap_{ijl} = \mu_{CAP}(a_i, s_j, o_l)$, $pass_{ijkl} = \mu_{PASS}(a_i, s_j, s_k, o_l)$, and $perm_{ijkl} = \mu_{PERM}(a_i, s_j, s_k, o_l)$.

For all actions, subjects and objects, we want to modify the relations using the following rules.

for CAP: $cap_{ikl} :=^2 cap_{ikl} \vee (pass_{ijkl} \wedge cap_{ijl})$

for $PASS$: $pass_{ikml} := pass_{ikml} \vee (perm_{ijkl} \wedge pass_{ijml})$

Since the transfers of capabilities and passes are performed between subjects, we can compute the modified CAP and $PASS$ relations by computing on subrelations indexed

[2]is an assignment operator that assigns the value computed by the left-hand side to the variable on the right-hand side.

by actions and objects of the whole relations. We will have the following subrelations.

CAP_{a_i,o_l}[3] is
"For an action a_i on an object o_l, a subject _ *can* perform the action, to degree _."

$PASS_{a_i,o_l}$ is
"For an action a_i on an object o_l, a subject _ *can* pass to a subject _ a capability to perform the action, to degree _."

$PERM_{a_i,o_l}$ is
"For an action a_i on an object o_l, a subject _ *can* transfer to a subject _ the privilege of passing a capability to perform the action, to degree _."

Now the inverse of $PASS_{a_i,o_l}$ and $PERM_{a_i,o_l}$ would have the meaning of a subject *can obtain* from another subject specified capabilities and passes respectively.

At this point a *copy* operator, $\star$ is introduced to operate on two relations, R which stands for *receiving* relation and H which stands for *holding* relation. This $\star$ operator is defined based on the semantics of modification rules above. In general, we will have the following relations.

R is "A participant (receiver) _ can receive privileges from a participant (holder) _, to degree _"

H is "A participant (holder) _ has a privilege _, to degree _"

H' is "A participant _ recently earns a privilege _, to degree _"

Let $H' = R \star H$, H' can be computed by the algorithm below.

1. $H' := O$ (a null relation)
2. for p_i in P loop
   for p_j in P loop
   $privilegesH'(p_i) := privilegesH'(p_i) \sqcup (R(p_i, p_j) \wedge privilegesH(p_j))$
   end loop
   end loop

For each single step of an application of the $\star$ operator, we can obtain

$CAP_{a_i,o_l} := CAP_{a_i,o_l} \sqcup (PASS^{-1}_{a_i,o_l} \star CAP_{a_i,o_l})$, and

[3] CAP_{a_i,o_l} has been reduced to a one place relation, i.e. a fuzzy subset of S.

$PASS_{a_i,o_l} := PASS_{a_i,o_l} \sqcup (PERM^{-1}_{a_i,o_l} \star PASS_{a_i,o_l})$.

If the worst case scenario is to be computed, the $\star$ operator need to be employed iteratively a finite number of times to compute the modified $PASS$ until the relation does not change and then CAP again util it is stable. There will be a point where further application of $\star$ operator will not change anything, since the operator monotonically expands the relation until the maximum closure is reached. We can then combine these subrelations which are indexed by actions and objects, back to original CAP and $PASS$. After having obtained the modified CAP and $PASS$, the potential direct violations can be computed using the definitions in Section 3. For the potential indirect violations case, techniques to compute them are currently being developed by the authors.

References

[1] W. Bandler & L.J. Kohout, "Application of Fuzzy logics to computer protection structure", *Proc. 9th Int. Symp. Multiple-Valued Logic*, IEEE, 1979, pp 200-207.

[2] W. Bandler & L.J. Kohout, "Semantics of implication operators and fuzzy relational products", *Int. J. Man-Machine Studies*, vol **12**, 1980, pp 89-116.

[3] W. Bandler & L.J. Kohout, "The use of the checklist paradigm in inference systems", in Prade, Henri and Negoita, Constantin V. (eds.), *Fuzzy Logics in Knowledge Engineering*, ch **7**, Verlag TÜV Rheinland, Köln, 1986, pp 95-111.

[4] G.S. Graham & P.J. Denning, "Protection–principles and practice", *Proceedings of Spring Joint Computer Conference*, vol **40**, AFIPS Press, New Jersey, 1972, pp 417-429.

[5] L.J. Kohout & W. Bandler, "Analysis of capability-based computer protection models by means of fuzzy logics", *Proc. 11th Int. Symp. Multiple-Valued Logic*, IEEE, 1981, pp 95-99.

[6] L.J. Kohout & W. Bandler, "Computer security systems: Fuzzy logics", in M.G. Singh et.al. (eds.), *Systems & Control encyclopedia*, Pergamon Press, Oxford, 1986, pp 741-743.

[7] L.J. Kohout & B.R. Gaines, "The Logic of Protection", *Lecture Notes in Computer Science* vol **34**, Springer Verlag, Berlin – New York, 1975, pp 736-751.

[8] L.J. Kohout & B.R. Gaines, "Protection as A General Systems Problem", *Int. J. General Systems*, vol **3**, 1976, pp 3-23.