Argument Games for Interactive Access Control

Guido Boella
Università di Torino
Italy
guido@di.unito.it

Joris Hulstijn
VU Amsterdam
The Netherlands
jhulstijn@feweb.vu.nl

Leendert van der Torre
CWI Amsterdam and TU Delft
The Netherlands
torre@cwi.nl

Abstract

We are interested in interactive access control to web services in virtual organizations. We discuss argument games in which the set of credentials requested by the service provider to access a service is established by means of an interaction between a client acting as a proponent and a server acting as an opponent.

1. Introduction

Koshutanski and Massacci [5] introduce *interactive access control*, a kind of access control in which the resource provider can request additional credentials to a resource consumer attempting to access a resource. This approach is motivated by cases where the required credentials are not known to the client, and extends traditional access control mechanisms in two ways. First, policy rules are declarative, which becomes necessary once managing access control for web services and virtual organizations becomes more complex [2]. Second, access control is not only based on the certification of identities, as for example in the X.509 protocol, but also on the *credentials* needed to access the service according to the declarative policy. More generally, the interaction among client and server can be modeled as a dialogue among autonomous agents in which parties discuss modalities until they reach an agreement. They may not only request credentials but also use arguments to support their case, negotiate which policy to apply, etc.

To support interaction among client and server, Koshutanski and Massacci [5] extend the trust management system with a logical reasoning engine, which makes it easier to reason about the details of an access request, and about the context in which access requests are evaluated. The trust management system uses a storage of credentials presented earlier, and use *abduction* to work out any missing credentials for the current request. Credentials are provided by the applicant and verified against the access policy, using deduction. If the credentials are insufficient in the current context, abduction is used to work out which set of credentials is missing. In [3] we consider the case in which further security objectives can arise affecting the subsequent negotiation, and replace the logical reasoning engine by an argumentation theory. For example, once an agent has gained access to a resource, requests for another resource may be denied to avoid potential conflicts of interest.

In this paper we raise the question how to develop a dialogue theory using a logical reasoning engine or an argumentation theory. We need a theory of interaction when this interaction becomes more complicated, for example when clients and servers have their own preferences and principles concerning which credential to disclose or require, and parties may be said to argue about the outcome of an access request. A formal theory of interaction can be used to analyze the dialogue system, or simulate it. As an illustration, consider the following dialogue between a client A and a library clerk B at an automated online article repository.

A: I would like to retrieve this article here.

B: Yes, but you need a subscription.

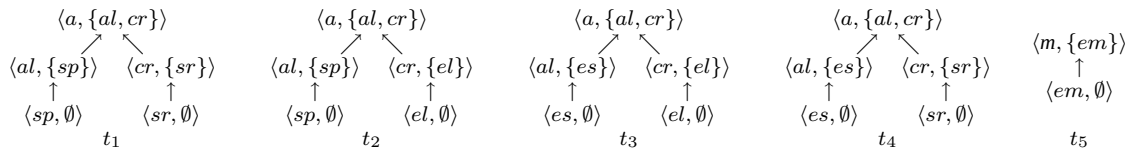
A: I am a University employee.

B: Please show me your pass.

A: $<$ showing pass $>$

A dialogue theory able to formalize and reason about such dialogues contains at least a set of agents, a set of dialogue acts, protocol rules defining the allowed dialogue acts in a state, and the logical reasoning engine or the argumentation theory. The straightforward approach to define such a theory of interaction is to define a finite state machine and use related techniques developed in theoretical computer science. However, the drawback of such an approach is that the theory of interaction is not related to the theory of argumentation. In this paper we consider argument games as developed by Vreeswijk and Prakken [8] in the context of argumentation theory.

The layout of this paper is as follows. In Section 2 we repeat the basic definitions of policies and objectives [3], and in Section 3 we define argument games for them.



2. Policies and Objectives

In this section we repeat the definitions of policies and objectives introduced in [3]. We distinguish among credentials (C) treated like atomic actions, and state variables (S).

Definition 1 Let C and S be two disjoint sets of credentials and state variables respectively. Let L be a propositional language built from $C \cup S$. A literal l is an element of C or S, or its negation. A rule is an ordered nonempty finite list of literals: $l_1 \wedge l_2 \wedge \ldots \wedge l_{n-1} \rightarrow l_n$. We call $l_1 \wedge l_2 \wedge \ldots \wedge l_{n-1}$ the body of the rule, and l_n the head. If $n = 1$ the body is empty and we write l_n. The *closure* of a set of rules R over a set of literals V, is defined by $Cl(R, V) = \bigcup_{i=0}^{i=\infty} S^i$ with $S^0 = V$ and $S^{i+1} = S^i \cup \{l \mid l_1 \wedge ... \wedge l_n \rightarrow l \in R,\ \{l_1 ... l_n\} \subseteq S^i\}$.

Definition 2 An *objective-policy description* (OPD) is a tuple $\langle O, P, K\rangle$ with O, P and K sets of rules from L, such that the heads of rules in P are built from a variable in S.

We use these rules to construct arguments or 'policies' that fulfill objectives, while no integrity constraints are violated.

Example 1 Let $C = \{$*es, sp, el, sr, em, ra, f, rm, ds, sm*$\}$ and $S = \{$*a, al, cr, m, cs, ib* $\}$, and consider the OPD:

$O =$ $\{ra \rightarrow a,\ a \rightarrow cs,\ rm \rightarrow$ *m*, *m*$\rightarrow ib\}$
$P =$ $\{al \wedge cr \rightarrow a,\ sp \rightarrow al,$ *es*$\rightarrow al,\ sr \rightarrow cr,$
$el \rightarrow cr,\ f \rightarrow cs, em \rightarrow$ *m*, $sm \rightarrow ib,\ ds \rightarrow ib\}$
$K =$ $\{ra,\ rm,\ em \rightarrow \neg es,\ em \rightarrow \neg el\}$

The goal is to grant access to an article (a) or a mp3 file (m), when requested ($ra \rightarrow a, rm \rightarrow m$). There are several ways to achieve both objectives, but if we add $em \rightarrow \neg sr$ as a third rule to K, there is no way to achieve both objectives. When granting access to an article the system wants to collect a survey ($ra \rightarrow cs$), and for each mp3 it requires improved bandwidth ($m \rightarrow ib$), *etc.*

A goal set is an option that is selected to be enforced, and which is derived from a set of *related* objectives, such that we can find mutually compatible policies that can realize them. Intuitively, a candidate goal argument is a goal argument, if its goal set cannot be split into a set of goal sets.

Definition 3 Let $\langle O, P, K\rangle$ be an objective-policy description. A *goal set* G is a set of literals. A *candidate goal argument* for *goal set* G, written $c(G)$, is a finite linear tree consisting of pairs of sets of literals with its unique leave (B, G) or any B, such that for each node (B, H) there exists an objective $l_1 \wedge \ldots \wedge l_n \rightarrow l \in O$ such that:

(a) $B = \{l_1, ..., l_n\} \subseteq Cl(K, U)$, where U is the union of all literals occurring in the ancestors of (B, H).

(b) if (B, H) is the root, then $H = \{l\}$, otherwise $H = \{l\} \cup H'$ when the unique parent of (B, H) is (B', H') for some B'.

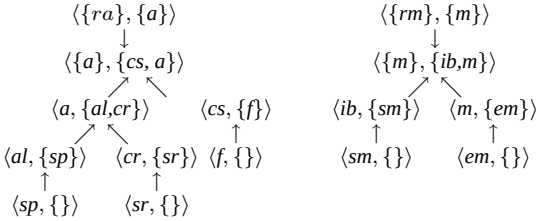
A *goal argument* for *goal set* G, written $g(G)$, is a candidate goal argument $c(G)$ such that there is no set of goal sets $\{G_1, ..., G_n\}$ with each $G_i \neq G$ and $G = G_1 \cup ... \cup G_n$. A *maximal goal set* is a goal set which has a goal argument and which is maximal with respect to set inclusion. We say that two goal arguments *conflict* if they contain nodes $\langle B_1, H_1\rangle$ and $\langle B_2, H_2\rangle$ such that $Cl(K \cup B1 \cup H_1 \cup B2 \cup H_2, \emptyset) \vdash \perp$, where $\perp$ stands for any contradiction.

Example 2 (Continued) The goal sets are $\{a\}$, $\{a, cs\}$, $\{m\}$ and $\{m, ib\}$. The set {*a, cs, m, ib*} is not a proper goal set, because it can be split in $\{a, cs\}$ and $\{m, ib\}$.

Plan arguments are constructed from policies, serving as a way to represent and reason with policies and their objectives. Some examples are visualized at the top of this page.

Definition 4 A *plan argument* for $\langle O, P, K\rangle$ for a goal in a goal set $g \in G$, written $t(g)$, is a finite tree whose nodes are pairs of a literal and a set of literals, either $\langle h, \emptyset\rangle$ for any $h \in C$, called a credential; or $\langle h, \{l_1, ..., l_n\}\rangle$ for any rule $l_1 \wedge ... \wedge l_n \rightarrow h \in P \cup K$, such that $\langle g, H\rangle$ is the root of the tree, for some H, $\langle h, \{l_1, \ldots, l_n\}\rangle$ has exactly n children $\langle l_1, H_1\rangle$, ..., $\langle l_n, H_n\rangle$, and the leaves of the tree are credentials. We say that two plan arguments conflict if they contain nodes $\langle h_1, H_1\rangle$ and $\langle h_2, H_2\rangle$ such that $Cl(K \cup H_1 \cup H_2, \{h_1, h_2\}) \vdash \perp$.

Goal-plan argument combines the two in the obvious way, as illustrated by two goal-plan arguments below.



3. Argument games

Argument games are dialogues between a proponent (PRO) and an opponent (CON). The main idea of interpreting interactive access control as a dialogue game is that the client is a proponent and the server is an opponent in an argument game. Vreeswijk and Prakken define argument games as follows [8].

Definition 5 (Vreeswijk and Prakken)

- A move is an simply an argument (if the first move) or else an argument attacking one of the previous arguments of the other player.
- Both parties can backtrack.
- An eo ipso (meaning: "you said it yourself") is a move that uses a previous non-backtracked argument of the other player.
- A block is a move that places the other player in a position in which he cannot move.
- A two party immediate response dispute (TPI-dispute) is a dispute in which both parties are allowed to repeat PRO, in which PRO is not allowed to repeat CON, and in which CON is allowed to repeat CON iff the second use is in a different line of the dispute. CON wins if he does an eo ipso or blocks PRO. Otherwise, PRO wins.

We thus have to define a set of arguments with a binary relation that represents which arguments attack which other arguments, i.e., an argumentation framework [4]. The attack relation is derived from conflicts between arguments [1].

Definition 6 An argumentation framework $\langle T, Attack\rangle$ for a objective-policy description $\langle O, P, K\rangle$ is an argumentation framework in which T contains all pairs $\langle t(G), l\rangle$ such that $t(G)$ is a goal-plan argument and $l \in G$. Let $S \subseteq T$ and $t, t_1, t_2 \in T$ be (sets of) such pairs.

- $\langle t_1(G_1), l_1\rangle$ attacks $\langle t_2(G_2), l_2\rangle$, iff either
  1. there exist two nodes p_1 and p_2 in the goal-plan arguments of t_1 and t_2 respectively, such that p_1 and p_2 conflict, or
  2. $t_1 \neq t_2$ and $l_2 \in G_1$: the literal of t_2 occurs in t_1's goal-plan tree.

The trees labeled $t_1, ..., t_5$ on the top of the page, are arguments for Example 1. The trees are alternative ways of realizing objective a, and therefore attack each other.

To illustrate three issues of using argument games, we now consider a simplified example considering only the dialogue in the introduction. It is based on the OPD with $O = \{a\}$, $P = \{s \rightarrow a, u \rightarrow a\}$ and $K = \{p \rightarrow u\}$. The objective is to get an article (a). Underlying the dialogue are two policy rules, to get an article you need to have a subscription (s), or be a university member (u), and an integrity constraint that university members have a pass (p). We have two plan arguments below, and the argumentation system $\langle\{T1, T2\}, \{attack(T1, T2)\}\rangle$.

$$\text{T1. } \begin{array}{c}\langle a, \{s\}\rangle \\ \uparrow \\ \langle s, \emptyset\rangle\end{array} \quad , \quad \text{T2. } \begin{array}{c}\langle a, \{u\}\rangle \\ \uparrow \\ \langle u, p\rangle \\ \uparrow \\ \langle p, \emptyset\rangle\end{array}$$

The first and most important issue is that dialogues are often modeled using speech acts like questions and answers, requests, etc, like the following pseudo-formal dialogue. In general, how to map dialogues to argument games is an open question.

PRO: I would like to retrieve this article here. $a?$

CON: Yes, but you need a subscription $s \rightarrow a, s?$

PRO: I am a University employee. $u \rightarrow a, u?$

CON: Please show me your pass. $p \rightarrow u, p?$

PRO: $<$ showing pass $>$ p

PRO: All right. Here it is. $p, p \rightarrow u, u, u \rightarrow a, a.$

The second issue is the fact that the two arguments are conflicting. In this example, there is no legal conflict between these arguments. It is quite possible to have a subscription and be a member of the university. But it is odd from a resources point of view. Why pay for a subscription, when you already have a membership? This motivates our attack relation.

The third issue is that dialogues often contain more than two participants, which is not covered by argument games. Consider the case in which we have several people requesting the same article. Since only one person can have the article, arguments with different people getting the article, will conflict. Usually, the person that gets there first, will get it. So the outcome in practice, depends on the way the dialogue develops. Suppose there were a dialogue protocol that forces people to speak in alphabetical order. In that case John would get the article ($j : a$), not Mary ($m : a$). Obviously, this example represents a whole class of exclusive access rights. The argument system is $\langle\{Tj, Tm\}, \{attack(Tj, Tm)\}\rangle$.

$O = \{j : a, m : a\}$
$P = \{x : u \rightarrow x : a\}$
$D = \{x : p \rightarrow x : u, x : a \leftrightarrow \neg y : a\}$, for any $x \neq y \in \{j, m\}$

$$\text{Tj. } \begin{array}{c}\langle j : a, \{j : u\}\rangle \\ \uparrow \\ \langle j : u, j : p\rangle \\ \uparrow \\ \langle j : p, \emptyset\rangle\end{array} \qquad \text{Tm. } \begin{array}{c}\langle m : a, \{m : u\}\rangle \\ \uparrow \\ \langle m : u, m : p\rangle \\ \uparrow \\ \langle m : p, \emptyset\rangle\end{array}$$

4. Credulous and sceptical reasoning

Prakken and Vreeswijk also show how their argument games are related to Dung's argumentation theory [4]. They show that an argument is in some preferred extension iff it can be defended in every TPI dispute, that is, if the dispute is won by PRO. Moreover, they show that in argument games where every preferred extension is also stable, an argument is in all preferred extensions iff it can be defended in every TPI dispute, and none of its attackers can be defended in every TPI dispute. In our example, we may have the following.

Credulous: both outcomes are possible. Which one is selected depends on the way the dialogue develops, e.g. on the order of requests.

Skeptical: the article is top secret. Only one person in the department can have access, to decrease the chances of leaking (avoid misuse). If two people claim access, there is something wrong, so no one will get access.

We have the same policy rules, but different meta-interpretations. There are many extensions in argumentation theory to deal with such cases, for example by adding rules that state that Mary has priority, because she is senior, in which case John argues that he needs the article for an emergency, etc.

5. Conclusion

There are two relevant kinds of reasoning in interactive access control, and we therefor distinguish two processes:

objective generation, the derivation of security objectives during the interaction, and

policy generation, the derivation of combinations of credentials that will achieve those security objectives.

Objective generation proceeds by forward reasoning from the current state of affairs to preferable objectives (deduction). In case several sets of mutually compatible objectives are derived, so called options, a selection has to be made on the basis of some priority principle. Policy generation on the other hand, needs backward or "means-ends" reasoning from objectives to required and missing credentials (abduction).

These types of reasoning can be related to planning. Policy generation is analogous to goal-based planning, and objective generation corresponds to goal generation in which goals are not given, but derived from, for example, desires and beliefs.

These two types of reasoning can also be formalized in an argumentation theory. An objective that has several possible policies or combinations of credentials, can be modeled just like an argument which consists of a claim with the supporting argumentations. The attack relation defined over the set of arguments can serve as a criterium to deal with possible conflicts between policies, and to select a set of compatible policies that achieves a set of objectives.

A general framework for interactive access control incorporating the two types of reasoning is dialogue theory, and we pioneered the use of argument games. A more general approach, for example based on Prakken's general framework [6, 7], can probably deal with the issues we found when formalizing a simple example.

An important issue to be studied is how the two kinds of reasoning, and roles of PRO and CON, can be realized in the reasoning engine of trust management system. PRO wants the applicant to have access, and CON tries to ban access. The open question is how the roles can maybe be mapped on tasks of the trust management system, such as the following.

task 1: construction. This happens when the applicant requests access, on the basis of a policy rule and some credentials.

task 2: criticism. This happens when conflicts are detected, and the request is evaluated against current access commitments.

References

[1] L. Amgoud and C. Cayrol. On the use of an ATMS for handling conflicting desires. In *Procs of KR'04*. AAAI, 2004.

[2] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *IEEE Symposium on Security and Privacy*, pages 164–173. IEEE, 1996.

[3] G. Boella, J. Hulstijn, and L. van der Torre. Argumentation for access control. In *Procs. of AI*IA05*, LNCS. Springer, 2005.

[4] P. M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[5] H. Koshutanski and F. Massacci. A system for interactive authorization for business processes for web services. In *Procs. of ICWE04*, LNCS 3140, pages 521–525. Springer Verlag, 2004.

[6] H. Prakken. On dialogue systems with speech acts, arguments, and counterarguments. In *Proceedings of JELIA'2000, The 7th European Workshop on Logic for Artificial Intelligence*, volume 1919 of *LNAI*, pages 224–238. Springer, 2000.

[7] H. Prakken. Relating protocols for dynamic dispute with logics for defeasible argumentation. *Synthese*, 127:187–219, 2001.

[8] G. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Proceedings of JELIA'2000, The 7th European Workshop on Logic for Artificial Intelligence*, volume 1919 of *LNAI*, pages 239–253. Springer, 2000.