

# Power and Permission in Security Systems

Babak Sadighi Firozabadi \*      Marek Sergot

Department of Computing  
Imperial College of Science, Technology and Medicine  
180 Queen's Gate, London SW7 2BZ, UK  
{bsf,mjs}@doc.ic.ac.uk

## 1 Introduction

It is a standard feature of all organisations that designated agents, usually when acting in specific roles, are empowered by the organisation to create specified kinds of states of affairs — as when, for instance, a priest declares a couple as married and thereby makes it so in the eye of the church, or when a head of department assigns one of his subordinates to a particular project, or when an owner transfers ownership, as opposed to mere physical possession, of an item to another entity. This feature of organisations is referred to variously as ‘(legal) power’, ‘(legal) competence’, or ‘(legal) capacity’. Jones and Sergot [JS96] use the term *institutionalised power* to emphasise that this is not a feature of legal systems alone but commonplace in all organisations. The neutral term ‘institution’ is used by them, and other authors, for any kind of formal or informal organisation.

The states of affairs created when a designated agent exercises an institutionalised power have conventional significance or meaning inside the institution, though not necessarily outside it. For example, ownership of an object, in contrast to its physical possession, is not something that can be observed, and it is possible that one institution recognises an instance of ownership whereas another institution does not. Searl [Sea69] distinguishes between what he called *institutional fact* and *brute fact*. An instance of ownership is an institutional fact; possession is a brute fact.

Policies are institutional facts. For example, the security policies of a system hold within that particular system and perhaps not in any other. As pointed out in [JS96], words such as *authorisation*, *right* and *privilege* have a wide variety of meanings in ordinary usage. They are frequently used in computer system applications and in computer security, but still not in any standard way. Sometimes by a word such as ‘right’ we mean a permission, sometimes an institutional power, sometimes a combination of the two, and sometimes something else.

In legal theory, the importance of the following distinction has long been recognised. There are three quite different notions:

1. the power to create an institutional fact;

---

\* Part of the work was done during visits to Swedish Institute of Computer Science.

2. the permission to exercise that power;
3. the practical ability (opportunity, know-how) to exercise that power.

The distinction between permission and institutional power is illustrated in [JS96] by quoting the following example from [Mak86].

...consider the case of a priest of a certain religion who does not have permission, according to instructions issued by the ecclesiastical authorities, to marry two people, only one of whom is of that religion, unless they both promise to bring up the children in that religion. He may nevertheless have the *power* to marry the couple even in the absence of such a promise, in the sense that if he goes ahead and performs the ceremony, it still counts as a valid act of marriage under the rules of the same church even though the priest may be subject to reprimand or more severe penalty for having performed it.

In this case the priest is *empowered* to marry a couple, which is an institutional fact, but at the same time he may not be permitted to do so. It is common, first to empower an agent to create a certain institutional fact, and then separately to impose restrictions on when that agent may exercise his power.

We sketch a simple formal language for expressing permission and power in security policies, and discuss briefly the significance of the distinction in connection with control mechanisms.

## 2 Detective and Preventative Control Mechanisms

Part of any security system is its control or monitoring mechanism, which has to ensure that agents are behaving according to the specified security policies. One can distinguish between two basic kinds of control mechanism, called in [FTL99] *preventative* and *detective* control mechanisms. A preventative control mechanism prevents an agent from violating the policies, whereas a detective control mechanism does not *guarantee* that violations are prevented, but will ensure that a violation is detected in some reasonable time.

The distinction between the two can be summarised by saying that a preventative control mechanism satisfies the following formal property, whereas a detective control mechanism does not.

$$\vdash \phi \rightarrow P\phi$$

Here  $\phi$  represents a proposition such as “*a* reads file *F*”.  $P$  stands for a standard deontic permission (see e.g. [Che80] for details on standard deontic logic).

As formulated here, the violation of a policy is modelled as logical inconsistency. It is the function of the control mechanism to prevent violations by blocking actions that would lead to inconsistency. It is possible to make finer distinctions between different kinds of preventative control mechanisms, for example by distinguishing between what is logically possible and what is practically possible, but we shall not do so here.

In contrast to preventative control mechanisms, in detective ones an agent may be able to perform some action or bring about some state of affairs, without having the permission for doing so. The following formula represents such situations.

$$\phi \wedge \neg P\phi$$

For many applications, the prevention of all non-permitted actions is not feasible or even desirable. For example, a bank may not want to impose practical restrictions on the amounts with which its brokers can trade, even if there are policies prescribing such limits. The managers may prefer to allow the possibility of violations, as long as there is a mechanism for detecting and recording them, for example as part of an audit trail.

### 3 A Formal Framework

The aim of our research is to develop a formal framework for representing and reasoning about access control policies and meta policies governing changes of these policies. The framework is also aimed to support modelling of control mechanisms to enforce specified policies and rules. We give here just a sketch of the main features.

We distinguish between two levels of policies and the control mechanisms for each. The first level contains access control policies specifying what actions agents are permitted and prohibited to perform on the various objects of the system. These access policies are represented using deontic operators for permission  $P$  and prohibition  $\neg P$ , as expressions of the form  $P\phi$  and  $\neg P\phi$  where  $\phi$  represents a proposition about actions of type “agent  $a$  reads file  $F$ ”, “agent  $a$  writes file  $F$ ”, and so on. Note that  $\phi$  in these expressions represents a brute fact, but  $P\phi$  represents an institutional fact.

The second level is the level of meta policies regulating changes of access control policies. We introduce a new (relativised) operator  $Pow$ , such that  $Pow_a \psi$  says “agent  $a$  is empowered to create the institutional fact  $\psi$ ”. The expression  $Pow_a \psi$  represents an institutional fact.

In this framework, the only means for changing policies are *declarations*, that is to say, illocutionary acts of *declarative* type in the sense of [Sea69, Van90]. An empowered agent declares that  $\psi$ , and by declaring it makes it so; the fact that  $\psi$  holds is due to the act of declaration.

We introduce a (relativised) operator  $Declares$  such that  $Declares_a \psi$  stands for “agent  $a$  declares that  $\psi$ ”. In such expressions  $\psi$  is a proposition representing an institutional fact — it is not meaningful to ‘declare’ a brute fact. The proposition  $Declares_a \phi$  itself, however, represents a brute fact.

Declarations are not necessarily successful (effective). The main relationship between  $Declares$  and  $Pow$  is the following:

$$[DECL] \vdash Declares_a \psi \wedge Pow_a \psi \rightarrow \psi$$

[DECL] expresses the *exercise* of a power to create  $\psi$  by designated agent  $a$ .

There are some similarities between our framework and the formal calculus for access control given in [ABLP92]. That calculus has two main components, a calculus of principals and a modal logic of principals and their statements. There are two (relativised) modal operators:

- $a$  Says  $\phi$ ,
- $a$  Controls  $\phi$ ,

The operator *Says* is defined as a normal modal logic (see [Che80] for properties of such logics). The operator *Controls* is defined in terms of *Says*, as follows<sup>1</sup>.

$$a \text{ Controls } \phi \stackrel{\text{def}}{=} (a \text{ Says } \phi) \rightarrow \phi$$

$a$  Controls  $\phi$  can thus be read as “agent  $a$  is believed on  $\phi$ ”, “agent  $a$  is reliable in regard to  $\phi$ ”, or, as suggested by ABLP, “agent  $a$  is trusted on  $\phi$ ”. In [ABLP92] each entry of an access control list (ACL) is given as a statement of the form  $a$  Controls  $\phi$ . As the authors explain, since such a statement records a server’s trust in  $a$  on  $\phi$ , if  $\phi$  represents a request (perhaps in an imperative form) to a server by  $a$ , then the server will grant it.

One half of the definition of *Controls* is:

$$a \text{ Says } \psi \wedge a \text{ Controls } \psi \rightarrow \psi$$

which has a clear structural similarity to [DECL] above. However the two logics are different. We are currently investigating how much of the ABLP calculus of principals can be incorporated usefully into our framework.

For each of the two levels of policies, access and meta policies, there is a need for a control mechanism to enforce them. In both cases, these control mechanisms can be of a detective or a preventative kind.

For the meta policies, the idea is that in a preventative system, the control mechanism blocks any declaration which, if effective, would violate a policy. Suppose, for example, that the owner  $a$  of a file  $F$  is empowered to permit another agent  $b$  to read  $F$ , represented by

$$Pow_a P(b \text{ reads } F)$$

and moreover to empower a third party  $c$  to permit  $b$  to read  $F$ :

$$Pow_a Pow_c P(b \text{ reads } F)$$

The formal property of preventative control systems in Section 2,

$$[\text{Prev1}] \vdash \phi \rightarrow P \phi$$

---

<sup>1</sup> This definition together with properties of *Says* as a normal modal operator has some very undesirable features e.g., the validity of  $a$  Controls  $\phi \wedge a$  Controls  $\psi \rightarrow a$  Controls  $(\phi \wedge \psi)$ .

where now  $\phi$  stands for any brute or institutional fact characterises a kind of preventative control system for meta level policies also. For suppose that we add to the example the following policy

$$\neg P(b \text{ reads } F)$$

If  $a$  attempts to exercise his power to permit  $b$  to read  $F$ , the declaration is blocked because  $\text{Declares}_a P(b \text{ reads } F)$  leads to inconsistency. However, a declaration by  $a$  that  $c$  is empowered to permit  $b$  to read  $F$ ,  $\text{Declares}_a \text{Pow}_c P(b \text{ reads } F)$ , is not blocked. But then an attempt by  $c$  to exercise *his* power,  $\text{Declares}_c P(b \text{ reads } F)$ , is again blocked because it leads to inconsistency.

In general, the property [Prev1] implies

$$[\text{Prev2}] \vdash \text{Declares}_x \psi \rightarrow (\text{Pow}_x \psi \rightarrow P\psi)$$

for any institutional fact  $\psi$ .

A system with property [Prev2] but without property [Prev1] is a preventative control system for meta level policies but not for access level policies. A slightly stronger preventative control system for meta level policies satisfies the following requirement:

$$[\text{Prev3}] \vdash \text{Pow}_x \psi \rightarrow P\psi \quad (\text{any institutional fact } \psi)$$

Of course it is not always desirable to have preventative control even for meta policies. In that case, in a detective system, the conjunction

$$\text{Pow}_a \psi \wedge \neg P\psi \wedge \text{Declares}_a \psi$$

can be consistent.

## 4 Delegation and Role

Delegation is discussed by a number of researchers as one of the issues in distributed systems and in particular access control mechanisms for such systems, see e.g. [Cri98,YS96]. Delegation is usually defined as giving certain *rights* to an agent to act on behalf of the delegator. However, the term ‘right’ in this context does not always mean permission, but sometimes institutional power. For example, a delegator issues a proxy to a delegatee to sign certain documents on his behalf. The ‘right’ that is delegated is not (merely) permission to sign the documents, but the power to create particular institutional facts by signing the documents on behalf of the delegator.

A delegation is a creation of a new policy e.g. a new permission or a new institutional power for the delegatee to act on behalf of the delegator. The act of delegating does not change any brute facts, but if successful, it changes some institutional facts. In our framework the mechanism for delegating is declaration.

There are policies in which a delegator is not permitted to delegate; a delegatee may not be permitted to do what is delegated to him; a delegatee may not be

permitted to exercise the power delegated to him, and many other possibilities. Our framework is able to express all these cases, and others.

A well-known method for organising and managing access control policies is by using *roles*. In role-based access control [SCFY96], a role is defined as an abstract entity relating a set of permissions to a set of users. In our approach a role relates not only a set of permissions, but also a set of powers to a group of users.

## 5 Further Work

We have sketched the main features of a formal framework and indicated how it may be used for expressing and reasoning about access control policies and meta level policies for changing policies. We are currently investigating whether incorporating features of the ABLP calculus of principals provides a treatment of groups, roles, and aspects of delegation. It is an interesting question whether the framework needs to be extended with other types of illocutionary acts such as assertion and request. It may be that the full significance will not become apparent until a temporal component is added to the framework. This is an extension on which we are currently working. Without the temporal extension we cannot distinguish between, for example, an agent's sharing of a power with another agent on the one hand and transferring it to the other agent on the other. We are also working on more detailed characterisations of what we called here preventative and detective control mechanisms, and the more general question of what it means to implement a security policy.

## References

- [ABLP92] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin. A calculus for access control in distributed systems. In Joan Feigenbaum, editor, *Proceedings of Advances in Cryptology (CRYPTO '91)*, volume 576 of *LNCS*, pages 1–23, Berlin, Germany, Aug. 1992. Springer.
- [Che80] B.F. Chellas. *Modal Logic - An Introduction*. Cambridge University Press, 1980.
- [Cri98] Bruno Crispo. Delegation of responsibility. In B. Christianson, B. Crispo, William Harbison, and M. Roe, editors, *Security Protocols*, number 1550 in *LNCS*, pages 118–124, Cambridge, UK, April 1998. Springer.
- [FTL99] B. Sadighi Firozabadi, Y.H. Tan, and R. M. Lee. Formal definitions of fraud. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems - New Studies in Deontic Logic and Computer Science*, pages 275–288. IOS Press, 1999.
- [JS96] A.J.I. Jones and M.J. Sergot. A formal characterisation of institutionalised power. *Journal IGPL*, 4(3):429–445, June 1996.
- [Mak86] D. Makinson. On the formal representation of right relations. *Journal of Philosophical Logic*, 15:403–425, 1986.
- [SCFY96] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role based access control models. *IEEE Computer*, 29(2):38–47, 1996.

- [Sea69] John R. Searl. *Speech Acts*. Cambridge University Press, Cambridge, 1969.
- [Van90] D. Vanderverken. On the unification of speech act theory and formal semantics. In Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors, *Intentions in Communication*, chapter 11, pages 195–220. The MIT Press, Cambridge, Massachusetts, 1990.
- [YS96] N. Yialelis and M. Sloman. A security framework supporting domain based access control in distributed systems. In *ISOC Symposium on Network and Distributed Systems Security (SNDSS96)*, pages 26 – 39, San Diego, California, Feb. 1996. IEEE Press.