New Security Paradigms: What Other Concepts Do We Need as Well?

John Dobson
Department of Computer Science
University of Newcastle upon Tyne
Newcastle NE1 7RU
United Kingdom

Abstract

Conventional approaches to computer security have concentrated on defining security in terms of access to resources implemented by locally imposed and managed constraints on simple access modes (e.g., read and write) to system resources (e.g., files and directories). It is now becoming accepted that this view of security is inadequate for managing security in a federation of administrative domains where local policies may conflict with global objectives and some negotiation is required to adjust multiple local policies in order to prevent local policy conflicts from hindering the achievement of a global policy. This new security requirement demands not so much new implementation technology as new concepts to be elaborated. We shall argue that issues of security policy need to be derived from understanding the way that responsibility and authority work in an enterprise, and that the conventional approach of giving priority to modelling resource protection in terms of subjects, objects and rules, formalising these in a 'security policy' and expecting the result automatically to achieve organisational security objectives, is to misunderstand any legitimate *local* agency the security system may have as a *global* agency.

1 A Perspective on Computer Security Modelling

1.1 Old Security Paradigms

There have been a number of important and influential milestones in the development of secure systems processing classified data. We will loosely refer to this application domain as military security. One of the most important milestones has been the realisation that it is possible to produce security models which are applicable to a wide class of secure systems. The seminal work in this area is that of Bell and LaPadula [Bell and LaPadula 1976].

Most military secure systems developed sine the late 1970's have been designed and built to the spirit of the Bell and LaPadula (BLP) model, if not to the letter. BLP has been beneficial in a number of ways. It has given a very clear requirement for system developers and evaluators. It has had a positive influence in ensuring that experience gained from developing one system could be applied to another. It has facilitated the development of formal tools for assessing security, and so on. However there have been problems with systems based on BLP, not the least of these being the discovery, in many supposedly secure computer systems, of covert channels: that is, means of communication which violate the security policy but were not foreseen in the security specification. In effect, the security specification was unable to support the stated security policy. This has not, hitherto, caused the foundation of BLP to be challenged, but it has caused work to be undertaken on refining the model.

Since the original papers by Bell and LaPadula there have been a number of attempts to produce more general models which take into account system properties such as covert channels. Two well known examples are the noninterference model of Goguen and Meseguer [Goguen and Meseguer 1982] and Sutherland's work based on possible worlds semantics [Sutherland 1986].

We can summarise both models by saying that they try to take into account information flow between two subjects no matter how it arises, whereas BLP is confined to constraints expressed in terms of components

*The use of the word 'paradigm' (as a noun) in the title is the only such use in this paper. Our understanding of the use of the word in a philosophical context leads us to prefer the phrase 'conceptual model' (e.g., of security) instead.

of the system state, such as files or directories. Nevertheless, despite the undoubted progress made by Goguen and Meseguer, Sutherland, and others, there still are considerable problems in building secure systems and verifying that they satisfy some stated security model. Dobson and McDermid [Dobson and McDermid 1989] have argued at length that these problems are *inherent* in the nature of the models so far chosen; that the problems could only be overcome by choosing more appropriate bases for the models; and they outlined a more appropriate (enterprise-oriented) basis for security models. This paper summarises the concepts embodied in the new security paradigs and explains some implications for secure system design and project management.

1.2 Current State of Computer Security

The following table, recently introduced at the Computer Security Foundations Workshop IV [LaPadula and Williams 1991], shows the amount of investigation that has been done in various aspects of computer security modelling. As the table shows, there has been a lot of work done on how to define the internal requirements and rules of operation of a secure system, based mainly on the models mentioned before. More recent work has examined security as a system property rather than as a property of a system component, thus allowing discussion of how to compose a system with a certain security property from a set of components with known properties; this is indicated by the indication of elaboration of system objectives and design. But, as LaPadula and Williams point out, there is still much work to be done in understanding what 'security means to the enterprise (as opposed to what property of a system is meant by the term 'secure'). In particular, the problem of relating system objectives ('secure') to management or organisational objectives ('security') cannot be addressed purely in terms local to the description of the system, such as components of system state.

In terms of this taxonomy, our work addresses in some detail the first stage of elaboration, which specifies what is to be achieved by an information-processing enterprise, an important component of which is a secure computing system.

In this paper we shall discuss a number of other issues that we have to understand before we can begin to create adequate new conceptual models of security which take account of the structure of the enterprise. Specifically, the concepts of responsibility and obligation, causation and consequence, authorisation, conversation or exchange (of valuable resources) all need to be examined before the abstract concept of security can be characterised; and, in addition, the idea of information needs to be analysed before the concept of information security can. In this short paper it is not of course possible to do justice to all these difficult concepts; we shall merely indicate what seem to be the most relevant features and supply references to papers which take the various matters up in more detail.

We shall in the paper explain our terms by considering the following case of a security breach, which we take to be the paradigm case for our paper:

> A client entrusts her* money to a bank. An untrustworthy bank clerk who is entitled under proper authorisation to transfer money from one account to another makes an unauthorised transfer of the money from the client's account to his own. Due to an oversight by the bank's internal auditor, this unauthorised transfer was never detected, and the client lost her money.

What seems to be important in this case is that the breach can be seen in terms not only of that aspect of security related to 'no unauthorised access,' but also that aspect related to 'no violation of duty of care' and that aspect related to 'no possibility of consequential loss'. Carefully defined use of terms could perhaps lead to these distinctions being made in terms such as 'security', 'trust' and 'safety' respectively; but all these terms are already overloaded, and in any case we would probably informally use the term security indiscriminately to refer to some mixture of them all when we characterise the bank as being 'insecure,' as it clearly is. It may be a side effect of the discussion outlined in this paper that the distinction hinted at above can be made clearer, but for the moment we shall assume that the term 'security' does indeed contain elements of all three aspects.

The rest of this paper is structured as follows. Section 2 discusses issues of causality and consequence, thus indicating the difference between 'no possibility of undesired behavior' (a causal notion) and 'no undesired behavior results in loss' (a consequential notion). Section 3 discusses obligations and responsibilities (which can be further divided into causal and consequential responsibilities), the idea being that where there is no 'duty of care to protect' involved, a security breach cannot be said to occur. Section 4 looks

*We shall throughout the paper use use the convention that the owner of a valuable resource is female and the attacker or security violator is male. This should resolve any ambiguity as to the reference of pronouns.

Table 1: Work performed in various aspects of computer security modeling

Stage of Elaboration	Traditional Emphasis	More Recent Modelling	Areas for Development
Enterprise Description			xxx
System Objectives		x	x
External Interface Requirements Model		x	x
Internal			
Requirements Model	xxx	xx	
Rules of Operation	xxx	xx	
Functional Design		x	x
Hardware/Software Specification			xxx

at authorisation and argues that key to authorisation is the idea of a conversation or exchange which brings into existence a set of obligations and responsibilities. This discussion leads to the theme in Section 5 that information and resources only make sense, and hence acquire value for security purposes, only when considered as media of exchange (in a very general sense). Section 6 indicates one view of the nature of information so that the question "And what in particular is an information security policy?" can be addressed. Finally, Section 7 indicates some implications for the management and design of an information security system incorporating multiple policies.

2 Causality and Consequence

A useful distinction which we have made before [Burns, McDermid and Dobson 1992, hereinafter referred to as BMD] is between safety and security in the following terms:

A *safe* system is one which cannot harm us *even if it fails.*

A *secure* system is one which cannot enable others to harm us *even if it fails.*

(Note that we say nothing about the nature of the harm; in many cases, this may include damage to, or exposure to risk of, something of value either to us or to our enemies rather than simply damage to ourselves.)

This distinction is expressed in simple causal terms: safety is defined in terms of the harmful action being the direct consequence of failure, whereas security is defined in terms of the consequence of failure being a (partial) cause of the harmful action. Although these definitions are a bit simplistic and in need of refinement, their emphasis on a causal difference is important. Another way of making the same point is to say that safety is a matter of no undesired failures, whereas security is a matter of no undesired faults, a fault being a cause and a failure a consequence (not all faults result in failures). In [BMD], we went on to examine further this causal difference between safety and security defined as properties of systems. Here however, we want to explore first some other issues surrounding security as a system property.

The reason for not wanting directly to define security as a system property is that such definitions have the undesirable effect of shutting out considerations arising from the human activity system surrounding the system with respect to which security is being defined. Trying to define security solely in terms of (for example) *unauthorised access to* or *information flow within* a system makes it impossible to discuss the nature of authorisation (whether some particular access was authorised or not may be a matter for the legal or social systems to decide) or the definition of information (it is a moot point whether something can be counted as information if no means is available of interpreting it or even of deciding whether it has structure). Indeed, we believe that much of the dissatisfaction with current computer security models arises from just this intuition, that concepts and mechanisms relating to the protection of something of value cannot be discussed without taking into account the value holder and the nature of the evaluation—and these exist outside the protection system, not within it, and so cannot be discussed in terms internal to the protection

system.

Security policies derive from policy objectives. What is needed is to be able to state policy objectives and the derived security policies in a way that shows the links between them; then we may begin to understand what set of characteristics it is about the security policies that enables us to say that they are *security policies* (as opposed to safety policies or privacy policies, for example). The position we are taking here, and argued at some length in [BMD], is that the causal structure relating to failure modes is one such characteristic, failure being defined with reference to not achieving policy objectives. There may be other such characteristics of course; in particular, [BMD] also argue that the nature of the harm caused is another. The following section will argue that a third characteristic for distinguishing security policies from other possible kinds of policy relates to the structures of responsibility and obligation that exist in the human activity system.

3 Obligation and Responsibility

We shall begin by distinguishing between responsibility and obligation. Agents hold *responsibilities* for particular states of affairs that can be described using words such as 'profitability', 'safety', 'adequacy of service' etc. It is important to note the distinction between such states and *obligations* such as 'to make a profit', 'to take appropriate safety measures' and 'to run a service'. These obligations arise from the responsibilities, and are discharged by the performance of appropriate actions.

We define responsibility in terms of a basic relation between two agents: the *responsibility holder* is responsible to the *responsibility principal* for a state of affairs: the *responsibility target*. There seem to be two different kinds of responsibility, which we term 'consequential' and 'causal' responsibility.

An agent is said to hold **consequential responsibility** for a particular state of affairs if he may be blamed for that state, even although he is not necessarily involved in any actions affecting that state of affairs (the doctrine of 'ministerial responsibility').

Obligations arise from the consequential responsibilities held by an agent. Obligations are constraints on the choice of action and may require that the agent perform an action. The important point is that obligations may be passed from one agent to another, whereas consequential responsibility always remains with the responsibility holder. However when an obligation is passed from agent A to agent B a new consequential responsibility relationship may be generated where agent B is the holder of the new responsibility and agent A is the principal to whom B is responsible for the new responsibility target. (Agent A is also still the holder of his original responsibility.)

When an agent discharges an obligation by performing an action that agent is said to have **causal responsibility** for the action. Causal responsibility is created when and only when an action has been performed.

Taking the example of responsibility for security in a bank, we note that the bank directors have responsibility to the customers (and possibly to the State) **for** the security of their customers' money. Obligations **to** take security precautions, to audit the trustworthiness of employees and computer systems and suchlike arise from this responsibility. These obligations are passed to other employees such as bank clerks who perform actions such as 'receive cash', 'clear cheques' and 'transfer money between accounts' in order to discharge these obligations. New consequential responsibilities, associated with these obligations, may be vested in the bank clerks, who will be responsible to the directors for their actions. On performing an action a bank clerk acquires causal responsibility for that action.

It should be noted that, if the bank clerk has consequential responsibility for the outcome of his actions (or lack of action) he holds this responsibility as soon as he acquires the obligation to transfer monies according to the customer's instructions. In contrast causal responsibility is merely for the doing of the action and not for its outcome, and it only arises when the action has been performed.

Thus a significant property of obligations and responsibilities is the way that they are related to each other through an 'induces' relation. We are actively exploring further the modelling of relations between responsibilities based on this idea. From the point of view of creating new conceptual models of security, however, the main implication is that introducing a new model of security into an organisation will have the effect, not so much of *representing*, but of *changing* the network of obligations and responsibilities in (the security-related part of) the organisation. There is work to be done in determining how to predict the implications of such changes, which we are investigating as part of a CEC-funded project (ORDIT).

4 Authorisation

The ORDIT project used an elaboration of Figure 1 in a case study for an Italian bank which was concerned with changes in responsibility structures consequent upon the proposed introduction of new technology.† The management issue involved was one of loss of control: that by devolving more functionality to a distributed computing system, it was not clear how authorisation (e.g. for the opening of new accounts) could still be handled centrally given the bank's desire not to decentralise its responsibility structures. This lead to a study of authorisation procedures in the bank. In fact, one of the main reasons for modelling responsibilities and obligations relevant to a security system is indeed that it provides an account of authorisation. Essentially authorisation is the action which creates a network of responsibilities and obligations and as it were sets the machine in motion. We have elsewhere used this kind of modelling technique to investigate some of the issues involved in automating (part of) an authorisation function in a telecommunications application [Dobson and Martin 1992]. By asking questions about the (re)location of responsibilities and obligations among the various agents involved when an authorisation function is automated, it is possible to determine the boundaries of the automated system and thereby some of the requirements on it.

As already mentioned, the *product* of authorisation is a network of responsibilities and obligations. What is of equal importance for our present purpose is some important features of the *process*. Authority is by its very nature a relation between two agents, that is authorisation is created or mediated by a *conversational* process and we turn now to describe the concepts needed to give an account of a conversation.

5 Conversations

There are a number of important entities and relations surrounding the idea of a conversation which we have explained elsewhere, and which are summarised in the following diagram.

There are three main types of entity in the language: Information or Resource Structures, Conversations and Parties.

Information or Resource Structures In the case of information, these are conventional file or database structures. These structures therefore contain what is often referred to as "data." For convenience, we shall refer to this entity as "information," though strictly speaking we are in this context thinking of it as a structured set of containers rather than as an interpretation of the contents of the containers. Other kinds of resource can similarly be related to each other, e.g. through relations such as 'part of' or 'requires'. 'Represents' is a typical relation binding an information structure to a resource.

Conversations The role of conversations will be described later in this section.

Parties These correspond to the agents in an enterprise model such as that described in our previous work [Dobson and McDermid 1989]. We use the word 'party' rather than 'agent' because we are here working in what is sometimes referred to as the information projection rather than the enterprise projection. (For an explanation of these terms, and their use as languages, see for example [ANSA 1989].)

Between these three types of entity there are six possible relations:

Information /Resource - Information/Resource This is what is often referred to as the *conceptual schema.*

Party-Party The relation between parties is described in terms of the role relations between them, e.g. customer-supplier, client-server, colleague-colleague etc. The set of these relations we call the *contractual schema.*

Conversation-Conversation Conversations can refer to each other, and one conversation can be dependent on another. The set of relations and dependencies between conversations we call the *exchange schema.*

Party-Information/Resource Parties both *access* and *interpret* information. Rules can be expressed over which parties are allowed to access and interpret what information (as opposed to merely accessing it). For example, a secretary who reads a letter on behalf of a principal is certainly to be given access to the information; but in so far as the secretary is merely (so to speak) part of the access path, any interpretation by the secretary has no validity or authority as seen by the organisation.

†Only responsibilities concerning security are shown for clarity.

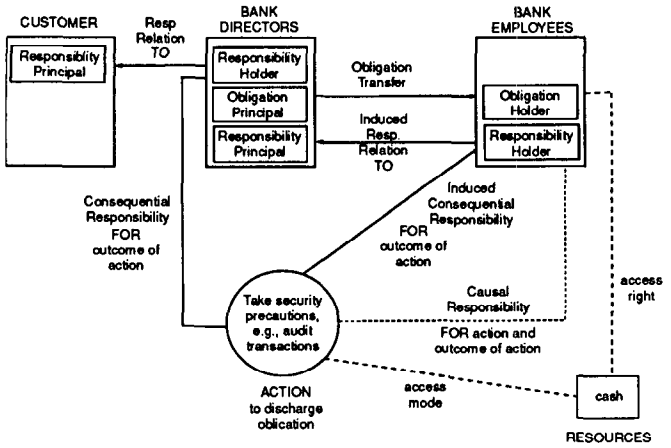

Figure 1: Example to illustrate the relationships between responsibilities and obligations

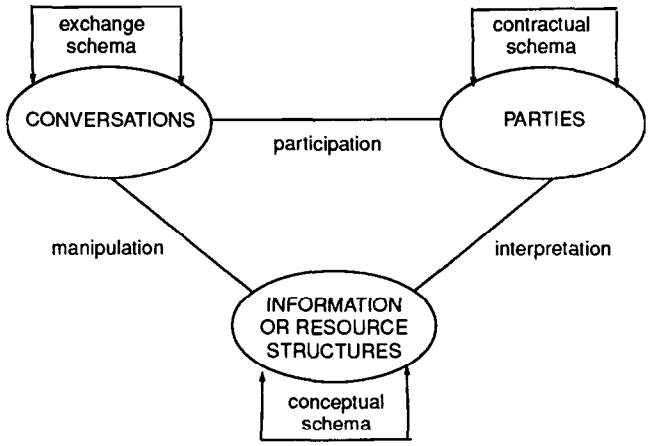

Figure 2: Basic entities and relations of a language for modelling conversations

Party-Conversation Parties *participate* in conversation. Rules can be expressed over which parties can participate in which conversations, and with what powers and authorities.

Conversation - Information/Resource Conversations *manipulate* information and information structures. (Manipulation includes the passive case of merely referring to information.) This means that information systems also participate in conversations, as when a secretary updates a database. This can be seen either as an abstract conversation between the secretary and a principal who later refers to the database, or as an abstract conversation between the secretary and the DBMS which would include such things as checks on the authority to access and update. Both views are, of course, equally valid; the relation between them is a relation between conversations and hence part of the exchange schema.

A very simple example will show how these concepts can describe all relevant aspects of information. We will suppose that John is employed by the University and is paid a salary of 24 *denarii* in 12 equal monthly installments. Simply recording this fact in a relational database in the form shown in Table 2 results in information loss: the table does not show whether the salary is paid to John or whether John pays the salary and in the latter case, to whom; nor is the fact of equal monthly instalments included.

We can compensate for this loss as follows. The table above is part of the conceptual schema. The fact that the salary is paid to John is part of the contractual schema relating John to the University, which is an instance of an employee-employer relation, and it is this role relation which describes the fact that, in return for certain obligations, John is paid a salary.

Table 2: Name-Salary relation

NAME	SALARY
John	24 (denarii)

The fact of equal monthly installments is (at least in principle) a result of an agreement reached during a conversation between John and the University's Finance Officer, a conversation which is related through the exchange schema with other conversations (such as the one between John and the University as a result of which an employment agreement was reached).

All of these components have to be described in an adequate information modelling language.[‡] It is not the job of this paper to define the syntax of such a language; but it is of interest to define what semantics must be capable of being represented by whatever syntactic structures are used by the information modelling techniques associated with any particular requirements capture language.

We note in passing that this set of concepts also allows us to talk about information privacy, in the following way. The basic concept of privacy seems to be this: that an individual can assume a number of roles in society and may well wish to maintain a separation between those roles. The right to privacy is then the right to have those separations respected by society (and its technological artefacts such as computers). Now the social notion of role is mapped on to our concept of a party to a conversation. We can thus talk about information privacy in such terms as mandating that a conversation which references data referring to an individual as a party corresponding to one social role (e.g. as a census respondent) cannot also reference that individual as a party corresponding to another social role (e.g., as a taxpayer). This is a constraint which would be expressed in the exchange schema. It is then possible to investigate denial of privacy, in the same sort of way as denial of service, as a violation of such rules. It is unfortunate that database technology does not permit the implementation of such concepts.

The previous sections have outlined some notions—causality, responsibilities and conversations—for which adequate conceptual models need to be built before an adequate conceptual model of security can be constructed, and we have attempted to outline the salient features of the required conceptual models. In the remainder of this paper, we shall similarly outline the main features of a conceptual model of security built on these foundations.

[‡]It is fair to say that, under this definition, we do not know of a single adequate information modelling language.

6 What is an Information Security Policy?

We have argued elsewhere [Dobson 1992] that there are four components of an information base:

- the **information structure model**;
- the **information rules**;
- the **information exchange specifications**;
- the **information rights**.

The **information structure model** defines the entities and their attributes. This corresponds to the straightforward use of a data model in database terms. Diagrams showing the information structure are especially useful, though they are not intended to be precise. Such diagrams are used in preliminary studies, and later for overviews. Any precise definition of the information structure uses a detailed information modelling language, such as INFOMOD [van Griethuysen and Jardine 1989]. The information structure model directs the design of databases or data files, particularly of subject databases (i.e., those centred around one entity-type). Each party has their own view of the information structure. As used here, the information structure subsumes both the (common) conceptual model and the (party-specific) external schemata.

The **information rules** define the interdependencies and conditions that must always hold among entities. These correspond to the integrity constraints that a DBMS might be expected to enforce plus other constraints that would be enforced by the database application. These rules control the integrity of the information. Integrity here means that the information is internally consistent. Integrity does not guarantee accuracy; that is, that the information truthfully describes the real world situation. Accuracy can be achieved only by management procedures (such as regular audits). However, if information is inconsistent, it cannot be accurate.

By 'semantic integrity' we mean the compliance of the database with constraints which are derived from

our knowledge about what is and what is not allowed or sensible in that part of the universe of discourse which is represented by data in the database. The maintenance of semantic integrity involves preventing data which represents a disallowed state of the universe from being entered into the database, or preventing a disallowed state of knowledge in the universe from being extracted from the database.

The maintenance of semantic integrity is currently a difficult and poorly understood subject. Semantic integrity constraints can be of varying degrees of complexity and the development of general purpose integrity checking languages, algorithms and implementations is a research issue of some interest and difficulty.

The **information exchange specifications** are part of the information model; they define the allowable manipulations of the information of the information and the specifications of message-handling processes, and are thought of in database terms as application-dependent specifications. Thus they specify the interface between (the users in) the environment and the information system. They describe the

- input and output messages to be exchanged between environment and information system;
- conditions and results of the actions manipulating the messages; the time-sequencing and coordination of the actions, indicating the order in which actions are carried out;
- authorisation rules, establishing which users may access or change which information.

The information exchange specifications specify *what* the information is to do. They do not specify *how* the system will do it, nor do they describe the forms in which the information exchange will take place.

The **information rights** define the rights of the various parties to the information structures, rules and exchange specifications, and are part of the access control mechanisms of the DBMS and its application. They answer questions such as "Who is the information owner[§]?" or "Who has the right to alter the rules and exchange specifications?" and so on. Note how the information rights differ from the information exchange specifications: the exchange specifications are extensional in logical form, whereas expression of the rights needs an intensional logic.

An important aspect of any enterprise is that operations on data are constrained by organisational structure and policy, and that some operations may be legal for some roles or individuals and illegal for others. Furthermore, the definition of what is legal should be open for amendment by some process or mechanism which is outside the data management system but reflected in it in order to implement the legislative changes.

The *information structure model* and *information rules* provide a common basis for understanding the universe of discourse. The *information exchange specifications* and the *information rights* define the allowed evolution and manipulation of the information. Together, these four parts specify what is expected of an information system.

We can relate these components of an information model to the conceptual model of conversations in the following way:

The **information structure model** and **information rules** together constitute the *conceptual schema.*

The **information exchange specifications** form part of the *exchange schema.*

The **information rights** form part of the *contractual schema.*

We have investigated the application of this conceptual model of information to the denial of service problem [Dobson 1992]. For example, each of the following represents a different kind of access restriction, and will have to be checked before an allegation of denial of service can be proved.

"The patient may not have access to the patient's own medical record."

This is a constraint on access expressed in terms of the information rights.

"Only the doctor can give the patient the right to access the patient's own record, and even then only after clearance from the hospital administrator."

This is a constraint on access expressed in terms of the information exchange.

"The patient must be currently registered (i.e. not discharged) if the medical record is to be made available to the patient."

This is a constraint on access expressed in terms of the information rules.

[§]Ownership may be taken as the right to destroy.

"The doctor's private notes do not form part of the patient record as seen by the patient."

This is a constraint on access expressed in terms of the information structure model.

The reason for wanting to characterise these as distinct types is that, in general, the constraints are expressed in different components of the application and its underlying database. Thus if an allegation (by the patient) of denial of service is made, it is necessary to check all four information sentence types to see if the allegation can be upheld. Similarly, if a prevention mechanism—or set of prevention mechanisms - is to be designed to prevent denial of access, those mechanisms need to take into account the fact that legitimate constraints of the above types are, in general, established and referred to in different types of conversations between different parties, and only by considering the whole set of information schemes can the prevention mechanisms be properly designed.

7 Implications

7.1 Implications for Project Management

Current security paradigms have concentrated on seeing security as an issue of centralisation: the notions of a security boundary, of a Trusted Computing Base, of access control as a primary mechanism, are all centralising concepts. But the problem of large distributed inter-organisational policies mean that the cosy assumptions of centralised administration, a single authority, and a security policy defined by an autonomous certification body, no longer apply.

Moreover, the next generation of PCs will be so powerful that it will be possible to divide a large application into pieces to run on the combined power of a network, using both distributed processing and distributed data. This raises the main issue for distributed computing as we now understand it: how can an organisation restructure itself so as to make most efficient use of a global computer network (as opposed to a network of individual computers). There are some security implications here similar to those arising from the considerations mentioned in the previous paragraph.

In the light of the situation outlined above, the following recommendations are suggested:

1. User organisations must tailor security to their own requirements, including those of cooperating and conflicting policies in a more or less loosely coupled federation. However, the problem of determining appropriate security requirements is not always easy, since the wrong requirements can act so as constrain the organisation's future. The danger is that it is easier to apply a model of security policy to an organisation's requirements than to determine the logical model of an organisation's security policy. The tradeoff between security and functionality is one that it is easy to get wrong.

2. The driving force behind security policies is to be understood at the level of the enterprise, not the level of the mechanism. This requires understanding the role of an architecture as a means of facilitating policy negotiation as well as directing an implementation (see [Zachman 1987] for a discussion of how to do this). In particular, security aspects must be considered at the architectural level.

3. User-based evaluation and certification require an organisational approach to risk management and assessment. Amongst other things, this means situating the user in the context of the organisation, rather than treating 'the user' as a single unity devoid of context (as is done in all current security models, for example).

4. Non-technical and social aspects such as organisational procedures and training must be seen as an essential component of the secure system and not as ancillary to it.

5. The contractual theory of information [Ciborra 1984] implies that information security is primarily a matter of protecting interests and only secondarily a matter of protecting information. This means that access control is not necessarily the only mechanism and that the value of information, both to its owner and to an attacker, has to be taken into account in evaluating countermeasures.

7.2 Implications for Multiple Security Policies

One of the major issues currently under discussion in the security community is that of *multiple security policies*: how can a global security policy be defined and managed in a system composed of a number of separate management domains, each with its own lo-

cal security management policy?¶ In many cases, the problems when they occur turn out to be ones of mismatch between management responsibility and management authority. Global responsibility should imply global authority; but security policy is often seen, and managed, as a matter of local authority. Analysis and resolution of this mismatch requires the kinds of concepts we outlined in Section 2.

To say that computer security is something to do with protection against unauthorised access to or manipulation of a valued resource may lead to the design of security *mechanisms* but will be of less use when applied to the statement of security policies, since these are derived from organisational objectives. In particular, security-related issues of interoperability and interworking must take account of possibly conflicting organisational objectives before possibly conflicting security requirements can be resolved, and this is a management problem to be solved at the management level using management concepts.

Thus the idea that the problem of multiple security policies can be solved simply at the level of the machine (e.g. by data labelling) is a mistake. Multiple policies means multiple policyholders, and the management problem to be solved is that of providing a framework within which the policyholders can agree mutual access constraints on resources that they wish to be shared. We are arguing in this paper that the proper expression of policy constraints in these circumstances requires not so much new machine implementations as new concepts to be implemented.

In general, multiple policyholders will wish to agree shared access only to particular resources and only for some particular purpose. (Since they are different policyholders they are presumably representatives of different enterprises, and therefore will have—at some stage—different interests which might well be in conflict with respect to at least some resources.) It is therefore important that a multi-policy machine be provided with the notion of a *context* of sharing, which will include amongst other things the common purpose which dictates the private resources to be shared. Without bringing in this notion of policy sharing in a context it is impossible to design the computerised security policy. The natural context of sharing is, of course, the conversation. We can define the shared context in terms of the conversations it contains.

¶This is a real question. It is said that during the Gulf War, problems were caused because although the USA and UK commanders were prepared to talk to each other, their computers were not because the computers' security policies conflicted. I have certainly known the same problem repeated on a much smaller scale within "collaborative" research projects.

Since the common purpose is a social rather than a technical consideration, this means that the problem of defining the shared and private components of the policy becomes one of drawing security policy boundaries in a socio-technical system. In order to model this drawing of boundaries, we have to decide on what things are in the model, with respect to which the boundaries are drawn; the fact that it is a socio-technical system means that we shall have to include some elements of enterprise modelling as well as security modelling.

We have found it convenient in our experience to draw these boundaries in a space containing responsibilities, conversations, agents (as parties to a conversation) and information and resources. In some work carried out in designing a privacy policy for a medical informatics application, we were able to express policy rules over access in these terms (e.g., "As a result of agreement between hospital administrator and ward consultant, patient may be given permission to see her medical record during or as a result of a particular consultation")—this might arise, for example, as a compromise policy between administrator ("Giving patient permission to see her medical record is a matter of hospital policy concerning confidentiality, which is my responsibility") and consultant ("Giving patient permission to see her medical record is a matter of medical judgement, which is my responsibility").

For these and similar reasons, it proved impossible in our work on defining security and privacy policies for the medical application to define the policies purely in terms of the entities and relations available in a standard relational DBMS. (See [Ting 1990] for an example of the kinds of security policy statements required. Our experience is that some real policy requirements can be even more difficult to formulate precisely, at least in terms of the concepts available in current security and database models.) We found we had to devise a more structured analysis of the concept of information, the result of which has been presented elsewhere [Dobson 1992].

7.3 Implications for Design

In this section, we shall indicate how a new conceptual model of security might draw together strands from these separate conceptual models of causality, responsibilities, conversations and information. We shall not define explicitly any particular security model, since the logical from of a security model can only be a construct derived from the requirements and organisational structure of the enterprise whose objectives it is intended to promote. We shall concentrate

on the *process* of constructing a security model suitable for a global organisational context of collaboration, rather than on its attributes as a product or logical construct. This process has a generic form which will now be described.

First, use a conceptual model of responsibilities to construct a network of responsibilities and obligations including, but not limited to, security-related actions and resources. The level of granularity here is a management decision: are responsibilities to be considered as vested in organisations, sub-organisational groups (e.g. departments) or individuals? This is needed to gain a clear picture of the authorisation structures needed to bring this network into operation, and may in addition allow discussion of enforcement mechanisms (e.g., the choice between enforcing in a machine, auditing in a machine, auditing in the human activity system).

Starting from responsibilities is crucial. Alternative starting points, such as activities or information (data) models far too frequently lead, in our experience, to conceptual models which, although they can be enforced by a machine, somehow seem to act against organisational objectives and are widely perceived by their users as restrictive or even oppressive rather than supportive, particularly in situations—and there are always are such situations—not foreseen by the designers.

Second, use a conceptual model of conversations to examine the conversations through which these obligations and responsibilities are established, invoked and discharged. (See [Dobson 1992] for more details of this.) Again, the chosen level of granularity is a management decision, but must clearly be related to the granularity level chosen in the previous step.

The two-sided nature of a conversation model is crucial. It is all too easy to overlook the fact that all actions have a cause and an effect, that information is a medium of exchange, that responsibility is held by someone to someone. An adequate security model must be able to represent fully this relational nature of its fundamental concepts. "Only authorised access to information"—*who* is authorised by *whom*? in the context of what *conversations* does the access take place? Between whom is the information *exchanged*?

Third, use a conceptual model of information to allocate different components of the security policy to different components of the information model.

The point of making the kind of distinctions we have made in our conceptual model of information [Dobson 1992] is that different components of the information base are accessed and manipulated during different conversations and by different parties. Concentrating on only one of the components (typically, the information structure model) leads to over-restrictive constraints on the kind of security policy that can be implemented. (Again, see [Ting 1990] for examples.) Widening the scope of the information model permits more generally applicable security policies to be designed and implemented.

Finally, avoid premature formalisation. In a sense, it is almost too easy to invent elegant new logical models of security—or of anything else, for that matter. It is also easier to *apply* a logical model of security to an organisation's requirements than it is to *determine* the logical model *of* an organisation's security requirements. The hard part of modelling is to answer the following questions:

- What are the organisation's objectives, its fundamental values, its criteria for evaluating its own failures? For example, a financial institution may prefer to insure against or recompense certain kinds of loss rather than prevent them, since its fundamental commitment is to financial trade-offs.
- How can the answer to the above questions be conceptualised, that is, what are the basic entities, attributes and relations in terms of which the conceptual models will be constructed? For example, creating a logic and proving an implementation is in conformance with it, is essentially a technique of fault prevention. However, an organisation that is structured to deal with insurance and compensation is essentially an organisation that values fault tolerance (forward error recovery) and any conceptual models that are formalised must be able to express concepts appropriate to an idealised fault-tolerant system; most logics fail this requirement. (See [Dobson and Randell 1986] for further discussion of the application of fault-tolerance to secure systems design.)

We have made a number of (unpublished) attempts to formalise our concepts of responsibility and conversations. They remain unpublished not just because the formalisms and logics were basically uninteresting and told us nothing we did not already know, but because the attempt led us too far away from trying to understand the way the concepts worked *in the context of the organisations we were studying.* Forcing a problem to be expressed in terms of its solution may sometimes produce a good solution to the problem as stated, but not usually to the problem. That is why

any new conceptual model of security must be largely based on a model of security as part of a process in the human activity system, not a model of security as an attribute of a technical system; and it is our belief that the concepts of causality, responsibility and conversation are basic to the kind of process model required.

Acknowledgements

Many people have contributed to the author's thinking on these topics over the years. It would be invidious to single out any subset of them, but they know who they are, and all are gratefully thanked. Whilst working in this area, the author has been funded by research grants from RSRE Malvern and the Commission of the European Communities (the ORDIT project, ESPRIT 2301), to both of whom proper acknowledgement is due.

References

[ANSA 1989] ANSA, ANSA Reference Manual, Release 01.00, Architecture Projects Managements Ltd., Cambridge, 1989.

[Bell and LaPadula 1976] D.E. Bell and L.J. LaPadula, Secure Computer System: Unified exposition and Multics interpretation, Report No. MTR-2997, MITRE, 1976.

[Burns, McDermid and Dobson 1992] A. Burns, J. McDermid and J. Dobson, "On the Meaning of Safety and Security," *Comp. J.*, vol. 35, no. 1, pp. 3–15, 1992.

[Ciborra 1984] C.U. Ciborra, "Management Information Systems: A Contractual View," in *Information Analysis: Selected Readings*, ed. R. Galliers, pp. 125–137, Addison-Wesley, 1984.

[Dobson and Martin 1992] J.E. Dobson and M.J. Martin, "Elicitation and Representation of a Security Policy for a Telecommunications Application," in *Eighth International Coference on Software Engineering for Telecommunications Systems and Services*, Florence, Italy, IEE, 1992.

[Dobson 1992] J.E. Dobson, "Information and Denial of Service," in *Database Security, V: Status and Prospects*, ed. C. E. Landwehr and S. Jajodia, Elsevier Science Publishers, Amsterdam, 1992.

[Dobson and McDermid 1989] J.E. Dobson and J.A. McDermid, "Security Models and Enterprise Models," in *Database Security, II: Status and Prospects*, ed. C. E. Landwehr, pp. 1-39, Elsevier Science Publishers, Amsterdam, 1989.

[Dobson and Randell 1986] J.E. Dobson and B. Randell, "Building Reliable Secure Systems out of Unreliable Insecure Components," in *Proc. Conf. on Security and Privacy*, Oakland, IEEE, 1986.

[Goguen and Meseguer 1982] J.A. Goguen and J. Meseguer, "Security Policies and Security Models," in *Proc. 1982 Symp. on Security and Privacy*, pp. 11–20, Oakland, CA, IEEE, 1982.

[LaPadula and Williams 1991] L.J. LaPadula and J.G. Williams, "Toward a Universal Integrity Model," in *Proceedings of the Computer Security Foundations Workshop IV*, pp. 216–218, Franconia, 1991.

[Sutherland 1986] D. Sutherland, "A Model of Information," in *Proc. 1986 Symp. on Security and Privacy*, IEEE, 1986.

[Ting 1990] T.C. Ting, "Application Information Security Semantics: A Case of Mental Health Delivery," in *Database Security, III: Status and Prospects*, ed. D. L. Spooner and C. Landwehr, pp. 1–12, North-Holland, Amsterdam, 1990.

[van Griethuysen and Jardine 1989] J.J. van Griethuysen and D.A. Jardine, Information Modelling with INFOMOD: Volume 1 - Concepts and Information Structure, Philips, Eindhoven, 1989.

[Zachman 1987] J.A. Zachman, "A Framework for Information System Architecture," *IBM Systems Journal*, vol. 26, no. 3, pp. 276-292,1987.