Messages, Communication, Information Security and Value

John Dobson

Department of Computing Science
University of Newcastle
Newcastle on Tyne NE1 7RU
United Kingdom

Abstract

This paper addresses the problem of analysing an information system for security flaws or vulnerabilities in a way that is analogous to the analysis of a safety-critical system. In particular, instead of adopting the approach that security is a property that must be proved to hold (fault avoidance), it shows how to analyse a system for possible security failures so that fault prevention, tolerance, recovery or even fault acceptance techniques can be chosen where appropriate.

1. Introduction

1.1 Value and Relevance

Information is one of the most dangerous substances known to humankind. Its use, or misuse, can bring down governments, destroy organisations, and cause untold personal misery. No wonder it needs to be handled safely and securely within a trusted security boundary.

In most approaches to secure computer information systems, however, it is common for the system boundary to be drawn quite tightly, so that computer security deals with failures (or provable lack of failures) in a well-defined physical system or well-defined program. Such a narrow view of 'the system' can be misleading when, say, the people interacting with the system form a significant source of failures according to a wider view. For example, it is sometimes the case when money vanishes through a computerised financial system to attribute the blame to 'employee fraud', when perhaps more insight would be gained by viewing the employee as a component in a wider system. The inappropriateness of a narrow view of system boundaries seems very marked in the case of computer security, where often it is precisely the fact that the attackers are interacting with the computer system and its human owners that defines the problem.

In this paper we shall not investigate in detail how information security system boundaries should be defined in practice; rather, we shall try to define a set of concepts that is intended to apply to practical security problems, no matter how widely the system boundary is drawn. This means that, in addition to obvious attacks on the computer system (e.g. entering password guesses), we must consider attacks upon the wider system (e.g. bribery, blackmail of personnel), and attacks which tamper with the system design before the operational system even exists (e.g. Trojan horses).

It is because of the tight system boundary view that typically computer security has been concerned with 'subjects' and 'objects'. In a wider view there are other sorts of animal as well. So we want some concepts which can relate to subjects and objects but are not constrained by any view that these are the only sorts of things that are of relevance to security.

Thus we cannot take the position that the only protection must be afforded by access control, authorisation and authentication, and other standard mechanisms of computer security. This is not just because these mechanisms are inappropriate for viewing the attacker as a component in a wider system, but more fundamentally because in a wider system, information has *value*, and this value is finite (however measured). Information is of value, both positive, as when it is used as an organisational resource to help an organisation achieve its goals, and negative, as when it is used detrimentally in the wrong hands. Protection must also allow for an appropriate balance of values; and protection adds value to information. This means that our view of security must explicitly take into account the value of information so as to enable tradeoffs to be made: for example, whether the value of some particular information is such as makes it worth protecting against all conceivable attacks, or only against some of them, e.g. those that will be cheapest for the attacker.

Another aspect of information is that of *relevance*. For example, information privacy (e.g. of medical records) is not just a matter of preventing information getting into the wrong hands; anonymity also allows the protection of individuals so they can do their job or fulfil their role better, whether as patient or doctor. The negative side of this is that it can also be used to cover up instances of medical malpractice or deception by the patient. These are both examples of the relevance, or the lack of it, of information. In the example given, names are not relevant to the medical use of the information, but would be relevant to claims of medical malpractice.

So any thinking about security to be used in a wider context must start from an understanding of the relevance of the information to what the organisation or relationship using the information is trying to achieve, produce a theory of information which is a value- and relevance-based theory, and see information protection as a value-adding or value-protecting process.

1.2 Security Management Policies

Very few, if any, current theories of information security are capable of reasoning about the value of the objects that the security policies are designed to protect. The usual assumption is that access to the objects is allowed, or not allowed, and that is the end of it. A corollary usually is that if access is to be prevented, it must be prevented at all costs, and hence the need for the provable correctness of an implementation of the access control policy.

But the "at all costs" assumption is unrealistic. In practice, an organisation has the choice of the following options:

- protect the object by reducing the vulnerabilities in the system and ensuring that the protection control system works (fault[1] **avoidance**)
- reduce the threat by containing the enemy so that the very possibility of exploitation of vulnerabilities is prevented (fault **prevention**)
- reduce the risk to exposure by so arranging things that a single attack cannot result in total loss (fault **tolerance**)
- ensure that if loss occurs, some recovery or compensation is available (fault **recovery**)
- accept the risk and hope for the best (fault **acceptance**).

Each of these options has an associated cost and risk. Risk management will consider these, and the value of the object to be protected, and the direct and consequential losses that might accrue, and the cost and effectiveness of countermeasures, and come up with a *security management policy*, which states how a fixed budget is to be allocated between the various fault management options identified above.

So although access control may be a good set of mechanisms for fault avoidance at least in certain cases, it is not the whole story. A new approach to security must enable reasoning about the costs and benefits of fault prevention, tolerance, recovery and acceptance as well. It is unlikely that controlling access to objects is a suitable conceptual basis for these in the way that it is for fault avoidance, because in none of these is access the real issue. What do seem to be the issues that need to be analysed is summarised in the following table:

Strategy	Issues for Analysis	Mechanisms
avoidance	what are the objects to be protected and who is allowed access to them? if access is granted or taken, what further information might be deduced?	access control information flow control
prevention	what other agents are there in the world and what is the disposition of their forces? what are their capabilities and budgets?	attack, i.e. causing faults in the enemy environment reducing the number of attackers or their budgets or capabilities
tolerance	where are the single points of failure and concentrations of value?	distribution (fragmentation and scattering) redundancy
recovery	what are the available options for forward and backward error recovery? for compensation?	insurance compensation
acceptance	what is the probability of loss and direct and consequential costs?	

[1] a fault here is a weakness in the system that might possibly result in a security breach.

1.3 Types of Security Analysis

There is not a single all-embracing concept which can do justice to the modelling and analysis of all these strategies and enable comparison between them. There are at least four different kinds of analysis that are involved:

- **vulnerability analysis:** what are the weaknesses in the system from a security point of view? (In conventional terms, this is analogous to fault analysis.) It proceeds by examining various abstract models of the system.
- **threat analysis:** what agents or events in the outside world could enable a vulnerability to be exploited so as to result in a loss? (In conventional terms, this is analogous to failure mode analysis.) It depends on knowledge, or hypotheses, about the domain in which the system is situated.
- **countermeasure analysis:** what countermeasures to vulnerabilities and threats are available, and what are their costs and effectiveness? This requires some real world information.
- **risk analysis:** is protection worth it if the countermeasure is costly and perhaps not very effective? What is therefore the nature of the security management policy?

Each of these kinds of analysis will require its own set of models, concepts and methods. We shall consider suitable abstract models of a system for vulnerability analysis and give some examples of domain knowledge to show how a threat analysis might be conducted. We do not consider countermeasure analysis because it depends on real world information which will be very situation dependent, nor risk analysis because it depends on the results of countermeasure analysis. The output of risk analysis is a security management policy, whose main features have already been outlined.

This paper addresses the problem of analysing an information system for security flaws or vulnerabilities in a way that is analogous to the analysis of a safety-critical system. In particular, instead of adopting the approach that security is a property that must be proved to hold (fault avoidance), it shows how to analyse a system for possible security failures so that fault prevention, tolerance, recovery or even fault acceptance techniques can be chosen where appropriate. The following subsection outlines the basis of the approach; subsequent sections of the paper provide more detailed explanation of the techniques.

1.4 Basis of the Approach

We start from two simple definitions, one of a safe system and one of a secure system:

A **safe** system is one that will not harm me or cause me loss, *even if it fails.*

A **secure** system is one that will not give others the means to harm me or cause me loss, *even if it fails.*

Of the many points that may be elaborated from these definitions, we wish to concentrate on four:

1) The failure modes of a system are at least as important as the normal operational modes, and need at least as much analysis. It is very striking how conventional approaches to safety case presentation concentrate on failure mode analysis in order to show how the safety mechanisms (or their alternates) will behave in the presence of failure, whereas this aspect seems lacking from security case presentation, which concentrates on showing that failures will not occur (or simply makes this assumption). If a security case amounts to saying "*This* is secure provided *that* is reliable" then there are further questions to be answered.

2) Safety is defined in terms of direct consequence, whereas security is defined in terms of indirect consequence: somebody ("my enemy") must receive something to my possible disadvantage.

3) Both safety and security are relative to a particular observer or stakeholder ("me").

4) "Loss" is a value term; it can be quantified in terms of some abstract value system (money, or peace of mind, or national security for example).

So any method of analysing a system for security vulnerabilities must be able to satisfy the following requirements:

1) It must be able to define, and recognise instances of, failure modes;

2) It must be able to accommodate the notion of someone receiving or obtaining something (this is what distinguishes a method of security analysis from methods of safety analysis, which do not have this requirement);

3) It must be able to accommodate the notion of relativity to a stakeholder;

4) It must be able to accommodate the notion of value.

2. The Conceptual Basis: Messages and Communication

The strategy we shall adopt is, in outline, as follows:

i) We shall define the abstract syntax of a message, and of a communication. A computer system will be seen as a means of enabling human

communication through the passing of messages. This deals with requirement 2) above. It also deals with requirement 3) since communication will be defined in terms of stakeholders.

ii) We shall provide a complete enumeration of possible failure modes of messages and communications. Security analysis then consists of attempting to identify all the instances of messages and communications in a system (this is the hard part!) and analysing the defined failure modes for each instance. This deals with requirement 1) above.

iii) We shall indicate an approach to attaching the notion of value to a communication so that standard methods of transaction chain analysis can be employed. This deals with requirement 4) above.

In this paper such a method of security analysis is proposed. It is based on modelling a computer system as a message-passing system, whose purpose is to facilitate human communication. We shall say that message-passing is the *perspective* of the model. The perspective of a model is what the model concentrates on representing.

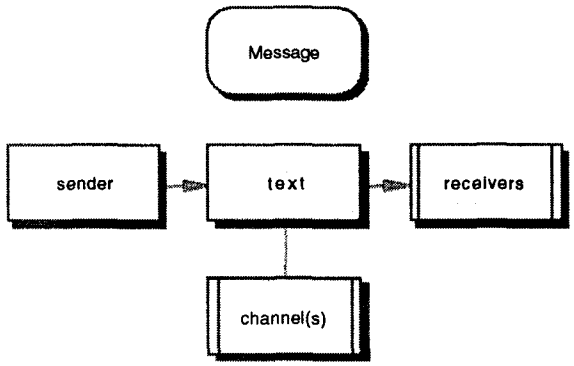
One of the main reasons for choosing messages as the perspective of our model is that it allows us to relate our notion of information to our notion of value. Information is what is passed via a message, and the message passing operation corresponds, in the case of an information commodity, to the value-adding operation described in the Section 1.1 of this paper. (This simple statement will be refined later, but for the moment it will do.)

Consider, for example, an information provision market, e.g. commercial databases. The database provider adds value by the making available of the information in the database. But a database can be considered as a message passing system. The sender is whoever puts the information into the database, and the receiver is the querent of the database; the database itself is the channel and the information the message text. Or consider in the military context a message sent from intelligence: the information increases in value as a result of successful transmission (not intercepted by the enemy and so on). Again, these simple explanations will be elaborated later in Section 2, when we distinguish between messages and communication.

For the moment, though, message passing is a value-adding operation over information; and the protection of information value is achieved through the protection of messages. Section 2.1 will now explain our model of messages in more detail.

2.1 Messages

The abstract definition of a message is: some **text** passed from a **sender** to a set of **receivers** over a **channel** (maybe more than one simultaneous channel, as in multi-media). No further elaboration of the primitive terms (in bold) will be provided here. We shall leave the definition of communication until later.



This definition allows us to enumerate the possible failure modes of a message:

- The apparent sender (as seen by a particular receiver) might not be the same as the real sender.
- The set of real receivers might not be the same as the set of receivers intended by the sender: some intended receivers might not receive the message and some unintended ones might.
- The text received by a particular receiver might differ from the text intended by the sender.
- There are a number of authorisation failure modes, all of them being some form of the sender not being authorised to send that text to a particular receiver.
- There are a number of sequencing errors over an ordered set of messages: message loss, message duplication, message permutation.
- The communication channel might block.
- The communication channel might suffer from a number of timing faults (messages delivered too late or too early).
- For a message whose text is delivered over a number of channels (e.g. multi-media), there is the possibility of synchronisation errors between the channels.

Our claim is that the above enumeration is complete, in the sense that any failure of an instance of a message (as defined) in a system can usefully be put into one of the above categories. The word 'usefully' implies that sometimes there may be a choice of which category to use.

2.2 Communication

Communication is a more subtle notion. The basic form of communication is an intention (i.e. a human interest, that which is to be communicated) being mapped by a particular stakeholder (the **speaker**) using a process we shall call **generation** onto a set of messages, which are then sent to a set of other stakeholders (**hearers**) who use individual processes of **interpretation** to reconstruct

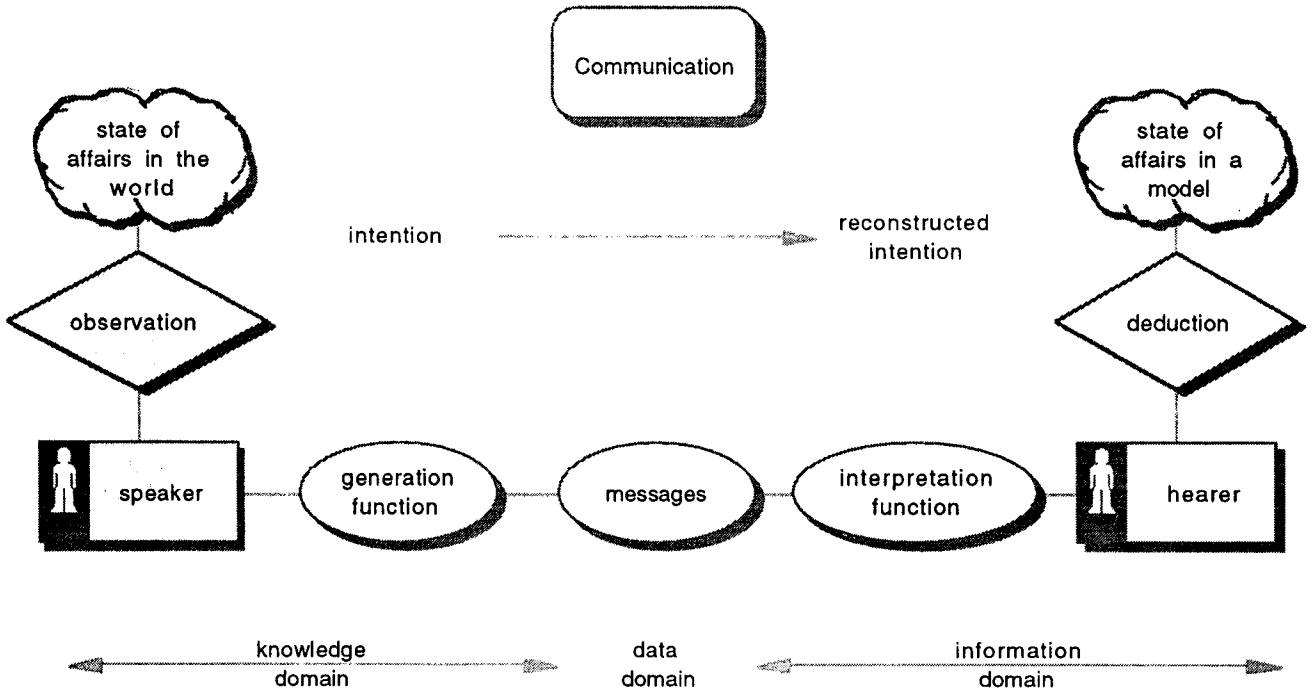
the original speaker's intention. Again no further elaboration of the terms in bold will be provided. For example, encryption can be considered one form of generation and decryption as the corresponding interpretation.

But there is more to communication than that, since we have to explain these unanalysed intentions. Our model is that the speaker's intention arises as a result of an **observation** of states of affairs in the speaker's world, and that a particular hearer's reconstructed intention results in the hearer adjusting the state of affairs in *the hearer's model of the speaker's world*. This model may be computational, or physical, or cognitive. Sometimes it may be the same as the speaker's world itself, but this is not always the case. We shall call this latter process of adjustment a **deduction** process.

The following figure is a summary of our communication model. Note that it defines three domains (knowledge, data, information), for later convenience.

This allows us to enumerate the possible failure modes of a communication, other than those that can be categorised as message failures:

- The reconstructed intention might not be the same as the original intention. Sometimes (but not always) this can be identified as a failure in generation (the messages do not carry the intention) or a failure in interpretation (the messages carry the intention but this does not get through to the hearer). Sometimes the generation and interpretation functions might not be mutual inverses.
- The original observation might be incorrect (not correspond to reality in the speaker's world).
- The intention might not capture the original observation correctly ("What I said was ... and this was mapped onto the messages correctly, but what I *meant* was ...").
- The deduction process might be faulty: the hearer makes inappropriate adjustments in the hearer's model.
- The hearer's model might be inappropriate: the hearer has chosen the wrong selection of state variables to select for representation, or to ignore, in constructing the model of the speaker's world.



We can now draw up a template which will be used for the categorisation of possible message failures, as follows:

	DATA DOMAIN FAILURES	
message failures	sequence failures	channel failures
real/apparent sender mismatch	*lost message*	*blocked channel*
real/intended receiver mismatch	*duplicate message*	*timing error*
sent/received text mismatch	*permutation error*	*synchronisation error*
authorisation errors		

Similarly we can set up a template of communication failure modes, as follows:

KNOWLEDGE DOMAIN FAILURES	INFORMATION DOMAIN FAILURES
observation errors	*deduction errors*
speaker intention errors	*hearer reconstruction errors*
generation errors	*interpretation errors*
generation not inverse of interpretation	*interpretation not inverse of generation*
	modelling errors

We can now define how we propose to carry out vulnerability and threat analysis.

Vulnerability analysis consists in identifying all the instances of messages and communications in the system in the terms we have outlined and using the templates above to identify the various failure modes for each instance.

Threat analysis consists in deciding whether each vulnerability is exploitable, given the domain knowledge or hypothesis that is being assumed. (A threat is always a threat with respect to an assumption.)

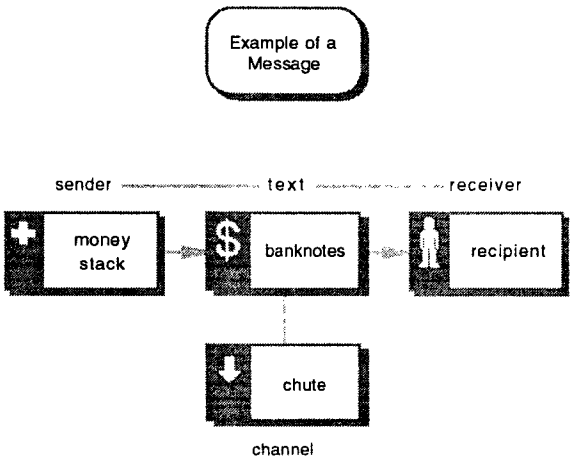
3. Two Examples

We shall take two very simple examples chosen especially to illustrate the ideas presented. Neither should be taken as examples of real world actuality.

3.1 Message Failure Example

Firstly, an example of a message:

At the final stage of an automated teller machine (ATM) transaction, some money is passed from a money stack through a chute to a grasping hand. This can be considered a message. The money stack is the sender, the chute the channel, the banknotes the text, and the grasping hand the receiver.



Applying our template, we can fill it in as follows. The numbers refer to subsequent paragraphs in which an example of possible vulnerabilities (security flaws) of that category is given. Threat analysis would decide whether such a vulnerability is exploitable.

(1) How does the recipient know that the money has actually come from the bank? This might not matter of course; but consider that a bogus ATM might have been set up in a shopping mall which dispenses real money in response to accepting, and stealing, clients' card numbers and identification codes.

(2) How does the bank know that the hand that grasps the money belongs to the person authenticated in previous messages? (It might not care of course.) Consider the opportunity for a thief who waits for a client to make a valid transaction and then grabs the money as it comes out of the chute.

(3) How do the bank and the client know that the money stack contains notes of the right denomination? Suppose the money stack contains grocery coupons? Suppose the chute tears all the banknotes in half as it delivers them?

(4) Questions of authorisation are supposed to have been dealt with during previous messages of this transaction.

(5) How do we know that the number of notes counted out by the money stack is the same as the number grasped by the hand? Is there a secret trapdoor in the chute that secretes every tenth banknote into a special cache for the benefit of the maintenance engineer, for example?

(6) Or does the money stack count "one for the client, one for the engineer, two for the client, two for the engineer..."?

(7) It probably doesn't matter in what order the notes are given out. But in other contexts, the exact sequence of messages might matter.

(8) This corresponds to the vandal who fills the mouth of the chute up with SuperGlue (™, probably).

(9) Suppose the chute only delivers the notes an hour after the money stack has been activated to deliver them. What would happen?

(10) In the case as given, there is only one

DATA DOMAIN FAILURES		
message failures	sequence failures	channel failures
real/apparent sender mismatch (1)	*lost message* (5)	*blocked channel* (8)
real/intended receiver mismatch (2)	*duplicate message* (6)	*timing error* (9)
sent/received text mismatch (3)	*permutation error* (7)	*synchronisation error* (10)

channel. If, however, we extend the model to include the paper record of the transaction that can also be given to the recipient, then this can be considered a multi-channel operation. Synchronisation demands that the money and the transaction record be given at the same time (with a suitable temporal granularity) for obvious security reasons.

What has been presented is a means of analysis, not a proposed new security protection policy. So what could be implemented is some automated support for the analysis. For example, in an object-oriented application, it would be useful to have records of messages passed to and from the objects of interest (the 'subjects' and 'objects' of the protection policy, for example), so that they could be analysed in the way proposed. What would also be of interest is to see how the messages related to the communications involved, assuming that there was some way of identifying or determining the latter. A prerequisite for this would be a decent enterprise model of the organisational environment in which the communication was taking place, since it is the enterprise model that allows definition of the types of communication that might take place.

3.2 Communication Failure Example

Now for an example of communication failure analysis.

> *The Prime Minister of Machiavellia suspects a plot among her colleagues to place her in a position of some political difficulty. She responds by secretly leaking a sensitive document to a journalist, so disguising things that it appears that the document has come from a colleague whom she wishes to embarrass first. (Leaking a sensitive document is, of course, a matter of embarrassment for the apparent leaker. Resignation will be demanded.)*

Here, the intention is clear — to embarrass a colleague. The messages generated and interpreted are of course, not just the sensitive document itself but more importantly the apparent source of the leak. Incidentally it is noteworthy that the security requirement on the document management system is not "It shall not permit security violations (i.e. leak documents)" but "Any properly authorised security violation shall have its authorisation concealed, and subsequently be auditable and traceable to any person the system owner nominates". There are a number of interesting implementations of such a requirement, none of which would pass normal security evaluation criteria. But it is still a proper security requirement from the system owner's point of view, the real purpose of the system being to protect her position, not documents.

(1) The Prime Minister might be in error in suspecting a plot.

(2) The document might not be sufficiently sensitive and embarrassing (its leaking must be a resignation issue), so it must be carefully chosen.

(3) The journalist might not understand that this is a deliberate leak (and, for example, return the document unread on the grounds that a simple mistake has been made).

(4) The Prime Minister must be assured that the source trail followed by the journalist, and any subsequent investigation of the leak, does in fact clearly point back to her colleague. Subverting (and possibly blackmailing) her colleague's Press Officer might do the trick — but again it might not. An apparent computer audit trail might be better if it could be arranged, but this might not be possible if the computer software is sufficiently "correct", i.e. not in accordance with the real requirements.

(5) The journalist might not recognise the document and the apparent source of the leak.

KNOWLEDGE DOMAIN FAILURES		INFORMATION DOMAIN FAILURES	
observation errors	(1)	*deduction errors*	(5)
speaker intention errors	(2)	*hearer reconstruction errors*	(6)
generation errors	(3)	*interpretation errors*	(7)
generation not inverse of interpretation	(4)	*interpretation not inverse of generation*	(8)
		modelling errors	(9)

(6) The journalist might not realise the apparent purpose for which the document has been leaked.

(7) The journalist might not believe that the apparent source is the true source (dangerous, this one!).

(8) The journalist might simply misread the identification of the apparent leaker.

(9) The journalist might not have an adequate model of Machiavellian cabinet politics.

It should be noted that it is precisely because the journalist in this scenario is a truly unpredictable individual who cannot be relied on to behave in any manner that is legal, moral, intelligent or committed that we want some way of analysing his possible failure modes. That is why we have taken a model of communication failures as a way of trying to decide what the journalist might try to do, or fail to do. For each of these failure modes, it is up to the speaker to decide whether to accept the risk or to reject it and find some way of avoiding, preventing, tolerating or recovering from it.

One issue of perhaps more than marginal interest here is how an individual develops the skills to be able to invent or conceive examples of each of the categories of failure mode. This is a matter of domain knowledge and there is probably no automatable short-cut. There are well-known social techniques for scenario generation, and using these is probably the best method that can be applied. What we have suggested is some way of providing some structure to help things along, as a way of systematically organising thought processes during scenario generation.

3.3 Summary

Perhaps some of the above assignments are a bit arbitrary, or could have been otherwise categorised. This does not really matter. What matters is that a systematic way is found of trying to generate as many different failure modes as can be conceived. To repeat, what we are advocating is a method which

- defines a model of messages and a model of communication;
- defines possible failure modes in terms of those models;
- instantiates the models wherever they can be found in the system to be investigated;
- for each instance, decides on suitable interpretations of the previously identified failure modes;
- for each interpretation of each failure mode, decides whether the security risk exposed should be prevented or the threat removed, masked (e.g. by insurance), or accepted.

It would be a mistake to think that the perspectives of messages and communication which we have presented are the only perspectives on an information security system from which a vulnerability and threat analysis could be performed according to the above process model. There are at least three other perspectives for which models need to be developed for a full security analysis of a system: *behaviour*, *structure*, and *enterprise relationships* (and possibly *substance* might be a fourth). In our previous work in this area, we suggested that from the point of view of analysis of security vulnerabilities, behaviour might best be modelled by some form of Petri net and structure was best considered in terms of a composition of trusted and untrusted components. We have also previously dealt with the enterprise relationship perspective at some length. But revisions of these, and an investigation of substance, might be for other papers.

Acknowledgements

Some of this paper is a revision and reworking of ideas and methods developed while the author was funded by RSRE Malvern (now known as DRA) during the period 1986-1989. Most of this paper has benefitted from close association over the years with Mike Martin. I am very grateful to the University of Connecticut for affording me the time and space to get these ideas down on paper in this revised and hopefully improved form and to T.C. Ting and Steve Demurjian for their helpful comments.