A Heuristic for Securing Hypertext Systems

Martin S Olivier

Department of Computer Science, Rand Afrikaans University

PO Box 524, Auckland Park, Johannesburg, 2006 South Africa

Email: molivier@rkw.rau.ac.za

Abstract

This paper considers the security requirements of hypertext and hypermedia systems. The emphasis is on the mechanism used to determine which subjects should be allowed to access which information. Many of the concerns are already being addressed by current secure database research—especially secure object-oriented database research. However, the particular structure of hypertext and hypermedia system allows one to study the particular requirements of such systems.

This paper considers the types of links that may be used in such systems and proposes a heuristic based on link types to simplify labelling entities to be protected. Such a heuristic is important since a typical hypertext system may contain extremely many nodes and links. The heuristic will help to reduce the burden of assigning labels to nodes and decrease the amount of time before the system can be used.

Keywords

Keyword Codes: H.5.1, K.6.5, H.3.5

Keywords: Multimedia Information Systems, Security and Protection, Online Information Systems

1 INTRODUCTION

The hypertext paradigm currently forms the basis of many computerised information systems. Hypertext is ‘*an assemblage of texts, images, and sounds—nodes—connected by electronic links so as to form a system whose existence is contingent upon the computer*’ (Slatin, 1991:56). Some authors prefer to emphasise that distributed computers have to be used and that the use of links between nodes are not the defining aspect of hypertext (Maurer, 1993):

Thus, we suggest that the term ‘hypermedia system’ is reserved for large, networked multimedia systems that support annotation, customization, and cooperation. The linkage aspect, sometimes extolled as THE defining aspect of hypermedia systems, seems to us of less importance, indeed may turn out to be as wrong an approach to structuring information as the goto has turned out to be the wrong approach to structuring programs.

Despite this comment, the current paper will indeed focus on the links between nodes—albeit a more structured view of links.

In many cases a distinction is made between hypertext and hypermedia, where the first term refers to systems that contain only textual information and the latter term combines various representations of information (including audio, video and text) (Feldman, 1994:69). In contrast, other authors use either term to refer to both types of system—see the definition above for example. In this paper we will use the term *hypertext* to refer to any system—whether it is text based or multimedia based. Only when we want to refer specifically to systems that use multimedia (and exclude text-only systems) will we use the term *hypermedia*. The term *hyperdocument* will be used to refer to the contents of a hypertext system.

Since hypertext is used to integrate related information it is logical that a hyperdocument that combines all related information, may include information not intended for all users of the computer system. It is therefore important to ensure secrecy of information by providing appropriate access controls. However, the number of nodes and number of links in a typical hypertext system makes this a formidable task. This paper presents a heuristic approach to facilitate this process.

The approach described in this paper is primarily applicable to commercial systems where security is important, but the high costs associated with absolute (military) security cannot be justified.

The paper is organised as follows: The next section contains an overview of relevant background material. Section 3 considers the use of secure object-oriented database technology to protect information in hypertext systems, while section 4 considers the specific security requirements of hypertext systems. Section 5 then describes the types of links that may occur in a hypertext system, followed by a section that describes cover stories—one of the mechanisms used in the approach described in this paper. Section 7 then describes the heuristic approach to access control in hypertext systems. This is followed by the conclusion.

2 BACKGROUND

A reasonable amount of research has been done regarding database security. Research regarding security in object-oriented and other ‘next generation’ database systems—see Olivier and Von Solms (1994) and Rabitti, Bertino, Kim and Woelk (1991) for examples—are particularly relevant for secure hypertext systems. An indication of how secure object-oriented databases may be used to protect information in an information retrieval system may be found in Thuraisingham (1993). Section 3 considers the use of such databases to store multimedia (and therefore hypermedia) documents.

Many hypertext systems are large distributed systems, often linking sites owned and administrated by different organisations. (As has been indicated in the previous section, some authors see this as a necessary condition for a system to be a hypertext system.) If the hypertext system is indeed distributed thus then it forms a prime example of a federated database and the recent research regarding such databases become relevant—see Olivier (1994), Olivier (1995), Pernul (1993) and Thuraisingham (1994) for examples. However, much work still needs to be done before secure federated databases will be a reality.

Security is generally divided into two categories: In discretionary security the owner of the information has the discretionary power to decide which other users may access information owned by this owner. Mechanisms such as access control lists are typically used to implement discretionary security; an access control list lists the identification of all users who may access an entity, as well as the rights they have when accessing the entity. In contrast to discretionary security, mandatory security is enforced on an organisational level and information flow is also restricted: A user who is allowed to access a piece of information A may not write it to a variable B where another user, who is not authorised to access A, may access the information in B. In most cases mandatory security is implemented with the multilevel security concept: every information item is assigned a sensitivity label; every user is assigned a clearance level. A user is then allowed to read an information item if the user's clearance level dominates ($\geq$) the sensitivity label of the information item. A user is allowed to write to an item if the item's level dominates that of the user (or, in some models, if it is equal to that of the user). Levels are often only partially ordered and usually forms a lattice (Sandhu, 1993).

The emphasis in this paper will be on discretionary security since discretionary security is currently the most widely used paradigm in commercial organisations—at least to some extent because it is more affordable. It is foreseen that access control lists will be the predominant mechanism used in hypertext systems as described in this paper. However, it is still possible to talk about 'more sensitive' and 'less sensitive' nodes even if discretionary security is used: if only some of the users who may access item A may access item B, then B is clearly more sensitive. As has been argued elsewhere (Olivier and Von Solms, 1994), viewed in this way, entities protected with access control lists (or similar mechanisms) also form a lattice. In this paper we will also use the term *security label*—a term usually reserved for the value indicating the sensitivity level in multilevel security; here it will refer to the access control list (or other protection) associated with a protected entity. When we say that node A is 'more sensitive' than node B we mean that the 'security label' of A is higher than that of B in the security lattice—that is, the labels of A and B can be compared and the label of A dominates that of B.

Even though the emphasis is on discretionary security, we do assume that protection for entities will be assigned by an individual (or group) in the organisation, known as the system security officer.

The next section considers the use of a secure database management system to protect multimedia documents. This is important since multimedia is one of the key technologies of hypermedia.

3 A TRUSTED MULTIMEDIA DATABASE

The use of object-oriented databases to store multimedia data is often cited as one of the major uses of such databases (Stein, 1994). This comes as no surprise since the variety of representation techniques inherent in multimedia makes it impossible to efficiently store in a rigorously structured database such as a relational database.

In addition to enabling a natural representation of multimedia objects, the use of objects also allows protection to be specified in terms of the particular medium: For example, assume that the concerned object is a voice recording. Such an object may normally provide methods such as `PLAYBACK`, `RECORD`, `VIEW` and `EDIT`, where `PLAYBACK` reproduces

the message via the speaker of the local computer, RECORD allows a new message to be recorded over the existing message, VIEW displays a graphical representation of the message and EDIT enables one to remove portions of the message using the graphical representation. Where most trusted relational databases will either allow any read operation or no read operation at all, it is possible in a trusted object-oriented database to distinguish between 'PLAYBACK' reading and 'VIEW' reading and to disallow none, any or both reading operations.

A reasonable number of models for secure object-oriented databases have been proposed: see Olivier and Von Solms (1994) for examples. All of these models may be used to store multimedia information securely and the differences between these models do not seem to point to particular requirements of multimedia security. In most cases the differences between multimedia and general object-oriented information seems to be at a different level than that of the proposed security models. For example, in a federated database the models proposed elsewhere (Olivier, 1994; Olivier, 1995; Pernul, 1993) may be used without change; however, the fact that it is known that multimedia is used may influence the choice of encryption protocols used for trusted communication between the sites.

Another difference between security for multimedia and for objects in general occurs when content and context based security measures are used. For example, it is relatively easy to build a security mechanism that allows a user X to access all employees' salary information, except if the salary exceeds \$100 000 (content based security). However, it is not currently feasible to build a security mechanism that allows user X to listen to recorded messages mentioning the salaries of employees, but denies X to listen to those where the salary exceeds \$100 000. Similarly, a security mechanism that allows X to view video clips of the office party, but denies access to clips in which the general manager of the company appears, is not feasible. Other potential (but not always feasible) possibilities include digitally transforming a recorded voice beyond recognition for some users, masking the face of an individual in a video clip for some users, removing the background scenes in video or background noise in audio clips for some users, and so on.

Context-based security considers the context in which information is to be used when determining whether access should be granted. Interesting possibilities in this area for multimedia security include the following:

- If the access request requires sound to be played back, and the workstation where the sound is to be played is situated in a (relatively) public location, the maximum volume can be limited. It may also be possible to limit playback of certain audio recordings to occur only on workstations that are equipped with earphones and not loudspeakers, or on workstations where it is possible to ensure that the sound is only played through earphones and not through the loudspeakers.
- Similar to the audio case above, replay of sensitive video material can be restricted to workstations that can ensure that the video cannot be relayed to an overhead projector or similar facility.

Despite the potential of possibilities mentioned in this section, they are not yet practical in most cases. In fact, most organisations will not use a secure object-oriented database currently, while *'means of developing MLS/FDS [multilevel secure federated database sys-*

tems] are generations away' (Thuraisingham, 1994). In what follows we therefore do not assume that a secure database is available.

4 HYPERTEXT SECURITY REQUIREMENTS

Let us assume that a new hypertext document is designed (possibly using some existing nodes) and is about to be entered into the system.

The first step is to determine who is allowed to access the various nodes—some nodes may contain information not intended to be accessed by all users of the system. These nodes have to be labelled accordingly. The heuristic approach to be described in section 7 will help with this process; initially we assume that the sensitive nodes are isolated by the system security officer and labelled appropriately.

However, labelling nodes may not be sufficient: Consider an experiment where the effectiveness of different weed killers are determined. Such a hypertext system will contain information about the different weed killers (active ingredient, manufacturer and so on); it will also contain information about the areas that are being sprayed by the weed killers (types of weed that currently occur at the site, other plants that occur, prevailing rainfall, prevailing wind conditions, etcetera). Additionally, a link will link each site to the weed killer being used at that site. Researchers may frequently access the hyperdocument to obtain information about the sites and to obtain information about the weed killers. However, for many experiment designs, most researchers should not be able to determine which weed killer is being used at which site before completion of the experiment. This means that it may be necessary to label links as well as nodes.

Labelling links may also not be sufficient: Consider a telephone directory where everyone can determine the telephone number and address of a given individual. However, for reasons of privacy, almost nobody is allowed to determine the name of an individual who lives at a specified address. The solution here seems to be to consider such a bi-directional link as two unidirectional links, where each can be labelled separately. (A similar view is expressed by Van Dyke Parunak, 1991, but for different reasons.)

Assume that all accesses to nodes occur via links. This assumption is realistic, since other access methods can be implemented with the aid of links: A directory or index can itself be seen as a node which is linked to the indexed nodes. A program that scans the document for a given piece of text (or other information) may traverse the document using links. Note that, if all links are labelled appropriately, it is not necessary to label nodes (although it will usually still be done).

To label links that have not been assigned explicit labels, the security label of the node pointed to by the link may be used. This means that, in this case, a given user may only access (traverse) the link if the user has the authority to access the node pointed to. Links that have been labelled, but are not at least as sensitive as the node it points to, will be relabelled with the least upper bound of the two concerned labels.

We will be using the assumption that a user is not supposed to know that a link exists if the user is not authorised to traverse the link. This means that, while the user is using the document none of the usual indications that such a link exists, should be given: the word that forms the starting point of a link should not be highlighted, neither should the cursor change when it is moved over the starting point of a link. We make this

assumption because hiding the existence of a protected entity from unauthorised users is a very effective first step in preventing that user from accessing the entity.

The described labelling suggests the following approach to displaying a node on a user's terminal (or playing it via the terminal's speakers or other facilities): Firstly, the node is composed in an appropriate manner in memory. The information about links available to the user is then read and inserted into the representation of the object. Note that in some cases the starting point of a link is anchored to a word (or picture element, or other structure) that inherently forms part of the node. In this case it is not possible to remove the word itself from the document—link information will only indicate that a link starts at the given word and adding the link information will highlight the word. However, in many cases the starting point of a link does not form an inherent part of a document: A 'button' labelled *More Information* below a piece of text can only be the starting point of a link and, therefore, such buttons and other indicators that are only used to anchor links, form part of the link representation and not of the node representation.

Thuraisingham (1995) also considers the labelling of nodes and links, as well as the issues surrounding multilevel nodes versus single level nodes.

We will now briefly consider some implementation issues.

It is common practice to build trusted database management systems 'on top' of trusted operating systems—the trusted operating system then forms a *Trusted Computing Base (TCB)* (Pfleege, 1989:339) and is trusted to enforce security. The role of the trusted database then becomes one of routing information to the appropriate operating system structure, where it will be protected by the operating system. The same approach can be followed for a trusted hypertext system. The link information can be written to files protected with the corresponding security label—that is, all links with security label S_1 will be written to a file with security label S_1 ; the same applies to links with security labels $S_2, S_3, S_4, \dots, S_n$.

If a trusted database system is available, it may offer more flexibility than the protected files mentioned in the previous paragraph.

Although links are the only mechanism provided to access nodes (and nodes therefore do not need separate protection), it is essential to store node information in files with security labels that correspond to those of the nodes to prevent access to node contents outside the hypertext system.

In this section we have assumed that it is possible to label nodes with a single label. However, that is not always the case. Database models that only support single level objects and models that support multilevel objects have been proposed and both types seem practical (Lunt, 1990). For simplicity we will therefore assume that single level nodes are used. See the conclusion for some remarks if multilevel nodes are to be supported.

The next section considers the use of links in hypertext systems in more detail.

5 LINKS IN HYPERTEXT SYSTEMS

One of the major problems that faces the designer of a hypertext system is to design it such that a user can easily navigate through it—and not get 'lost in hyperspace' (Gay and Mazur, 1991). Various mechanisms have been proposed to simplify the task of the user. One such approach, suggested by Van Dyke Parunak (1991), forms the basis of our security proposals. Van Dyke Parunak argues that the patterns used to order the

information graph and the types of link used between nodes, not only simplifies the task of a human user who navigates through a hyperdocument, but also improves the potential of automated searching.

The patterns described include a number of well-known graphs: linear graphs, cycles, hierarchies, grids (called hypercubes there), directed acyclic graphs and arbitrary graphs. The benefit of a regular graph for navigational purposes is that it is much easier for the user to keep track of the current position related to the larger system.

The link types depend on the purpose of a link: link types form a hierarchy as depicted in figure 1. At the highest level Van Dyke Parunak (1991) distinguishes between association links, aggregation links and revision links: Association links associate related nodes, aggregation links combine the various parts of a bigger node into a single node and revision links link previous (and possibly future) versions of a node to the current node. These links are again subdivided; as an example, association links are divided into word-proposition links and proposition-proposition links, where the former links a word to a sentence, phrase, node or other proposition and the latter links two propositions with one another. Word-proposition links are then subdivided again into links that link a word to its definition, links that link a word to further identifying information and links that link a word to a comment about the word. Figure 1 does not contain all link types identified by Van Dyke Parunak (1991) for reasons of space. However, all link types will be listed later (when we discuss figure 2).

Figure 1 Some link types identified by Van Dyke Parunak (1991)

6 COVER STORIES

Cover stories are ‘fictions intended to provide plausible, alternative explanations for observations that might otherwise lead to illegal inferences’ (Garvey and Lunt, 1991). As noted by Garvey and Lunt (1991), cover stories have to be consistent with other information in the database to be effective. We believe that the link types described in the previous section can be of significant help when designing plausible cover stories.

If the fact that a link starts at node N can be effectively hidden, then it is not necessary to provide a cover story on the other end of the link for unauthorised users. However, if it cannot be hidden then a cover story may be required. It is difficult to hide the fact that a link starts at a given node under the following circumstances:

- If all (or most) similar nodes are the origin of a link: For example, assume a company hypertext system contains a node for every employee, which is linked to a job description. If one employee does not have a job description then a logical (and probably correct) inference would be that it should not generally be known what the exact duties of this employee are.
- If a link historically existed: A manufacturing company may link a product to a description of the process used to manufacture the product. However, if the company finds a new process, not generally known, the company may want to update the database but not allow everyone access to the process description anymore.
- If a regular linkage pattern is used: If the nodes are arranged as a ring or a hypertorus (Van Dyke Parunak, 1991)—or any other regular pattern—it is easy to infer that a node is hidden if one is missing and conclusions can be drawn from this.

When designing a cover story the link types can describe the expectations a user will have when accessing the node: If it is intended to paraphrase another node, the cover story has to be brief and correspond to the main points given in the other node; if the cover story is intended to indicate the cause of the other node then it has to provide a plausible explanation for the other node. Obviously, it is possible that a cover story must simultaneously indicate the location for one node and serve as an illustration for another node.

7 SECURITY IMPLICATIONS OF LINKS

Figure 2 lists the link types identified by Van Dyke Parunak (1991) again. This figure also indicates whether the link can potentially point to a more sensitive node, whether the link can be hidden at its point of origin, and whether a cover story may be used at the other point of the link. When a user is able to access the node at the origin of a link, but not traverse the link then either the origin of the link should be hidden or a cover story should be provided in place of the real node pointed to. The table indicates whether the potential of a link to be more sensitive than the node at its origin is high (H), medium (M) or low (L). The same applies to the potential of the link to be hidden and the potential to use a cover story. However, the potential of a link to be hidden often depends on other factors (to be discussed) and this is indicated by an asterisk (H*).

To illustrate use of the table consider the *Content* (1.1.1) link type. It is a *Word-*

Level	Type	Sensitive	Hide	Cover Story
1	Association			
1.1	Word-proposition			
1.1.1	Content	L	H	L
1.1.2	Identification	M	H	L
1.1.3	Comment	M	H	H
1.2	Proposition-Proposition			
1.2.1	Orientation			
1.2.1.1	Location	M	H*	M
1.2.1.2	Temporal	M	H*	M
1.2.1.2	Circumstance	M	H*	M
1.2.2	Implication			
1.2.2.1	Causation	H	H*	H
1.2.2.2	Purpose	H	H*	H
1.2.2.3	Condition	M	H*	L
1.2.2.4	Contrafactual	M	H*	L
1.2.2.5	Concession	L	H*	M
1.2.2.6	Warning	M	H*	M
1.2.2.7	Evidence	M	H*	M
1.2.3	Paraphrase			
1.2.3.1	Amplification	H	H*	H
1.2.3.2	Summary	M	H*	M
1.2.3.3	Abstraction	H	H*	L
1.2.3.4	Instance	H	H*	L
1.2.3.5	Equivalence	L	H*	L
1.2.4	Illustration			
1.2.4.1	Manner	M	H*	M
1.2.4.2	Comparison	L	H*	L
1.2.4.3	Contrast	M	H*	H
2	Aggregation			
2.1	Argument	M	L	L
2.2	Discussion	M	H*	L
2.3	Legal case	M	M	M
2.4	Software module	M	L	L
3	Revision links	H	H	L
<hr/>				
Sensitive:	Pontential of link to be more sensitive than the node at its origin			
Hide:	Potential of link to be hidden from unauthorised users			
Cover Story:	Potential that a cover story may be substituted for the real node at the destination of a link			

H*: Generally with a high potential, but potential depends on the structure of the information graph.

Figure 2 Link types revisited

proposition link which means that it originates from a word in a node. In general, a reader of a document has to have a grasp of terms used in the document if the user is authorised to access the node. Therefore a *content* link that defines a word in the document will only be more sensitive than the word itself under unusual circumstances—this is indicated by the L(ow) potential increase in sensitivity for a content link. Further, if a word is the anchor for a link then it will usually be highlighted or otherwise marked as the start of the link. In general it should be possible not to mark it, thereby concealing the fact that the word is an anchor for a link (for some users). A content link is not a good candidate to point to a cover story: If a word is defined in one context with one definition and in another with a different definition inferences may be made. Worse still, in general it is expected that it would be difficult to find a convincing ‘alternative’ definition that fits all uses of the word throughout the hypertext system. This explains the L(ow) potential for cover stories linked with content links.

Consider *causation* links (1.2.2.1) as another example. In many cases the true cause of an action may be hidden from some users. For example, the reason behind improvements in a product may be similar improvements made by a competitor. However, if this fact is recorded in the manufacturer’s hypertext system, the manufacturer will prefer to tell most people that the improvements have been made ‘proactively to remain the market leader’. Obviously not all causes will be sensitive—the point is that a cause does have a relatively high potential to be more sensitive than the caused action (and causation links therefore have a H(igh) potential to be more sensitive than their origin node). In general causation links can be hidden easily: Causation links are proposition-proposition links and therefore either link a (relatively short) phrase or an entire node to another node. If only a phrase is linked, the same remarks that applied to linking words apply here. If the link links the entire node to another node the anchor will usually be a special ‘button’ (or other symbol). If this button or symbol is not displayed, the existence of the link is effectively hidden. However, if all events in a company’s hypertext system have linked causations then it will be difficult to hide a single causation link. Therefore the potential to hide the link has been marked with an asterisk in figure 2: like all similarly marked links it normally has a high potential to be hidden, except in the cases indicated in section 6. In general it is fairly easy to construct a ‘reason’ for an event—therefore causation can be pointed to a cover story relatively easily (that is, it has a H(igh) potential to be pointed to a cover story for some users).

Space does not allow us to give our reasons for classifying the various links the way we have done in figure 2. In addition, such a classification depends on the type of information that an organisation wishes to represent and the intended audience of the representation. It may therefore be necessary for particular organisations to adapt figure 2 to suit their circumstances.

We will now describe the approach based on figure 2 that may be used during the classification of nodes. Note that determining the type of a node is handled by the developers and is not primarily done to aid security—it is rather intended to support navigation through the system (Van Dyke Parunak, 1991).

Firstly, the potential for a link to be more sensitive than the node at its origin may be used to determine the order in which nodes and links will be labelled: those with a higher potential will be done before those with a lower potential. This may enable the information to be made available before all links and nodes have been labelled. This may be useful if availability of information is considered as relatively important when compared to secrecy

of information. Sorting of links according to potential sensitivity can be automated. Note that it may be necessary to re-sort links whenever a node or link is labelled.

After all (or the essential) nodes and links have been labelled, figure 2 may be used to decide between hiding links and providing cover stories. Here the potential to be hidden will be considered first: if the existence of the link can be hidden (an H and, possibly, an M in the *Hide* column) this is the recommended procedure. Otherwise a cover story will be recommended.

If a cover story is the recommended action, but the potential for a cover story is low, the node at the origin of the link also needs to be backed by a cover story such that any user who is supposed to see the cover story will also see the cover story at the origin of the link. This process needs to be performed recursively until all nodes and links are protected by being hidden or by having a suitable cover story available.

The identification of links to be hidden and nodes requiring a cover story can be automated, based on a table such as figure 2.

Generation of suitable cover stories obviously cannot be automated. However, knowing what the cover story is supposed to replace and knowing the expectation a user will have from the node (ie that it gives the cause of another node, an example of the node, etc) will significantly simplify the task of the system security officer who has to design suitable cover stories. Again, it is possible to design a tool (using links from the cover story to their origin) that will provide the required information to the system security officer.

8 CONCLUSION

The paper has addressed labelling of nodes and links in a hypertext system. A heuristic has been presented that simplifies the process of identifying the nodes and links of a given hyperdocument that needs to be protected. The heuristic approach also helps to identify cases where links cannot be hidden and cover stories may need to be used. In addition, taking note of the types of link that points to a node, simplifies the task of designing persuasive cover stories.

The paper assumed the use of single level nodes; if multilevel nodes are required, more work needs to be done. In particular, if a single word or phrase is more sensitive than the surrounding information, that word or phrase cannot, in general, be removed without affecting the entire node. Designing a cover story that corresponds to the original node in all important respects, but with a cover story concealing the word or phrase, may be substantially more difficult than designing a cover story where any information in the node may be changed. When such a node is updated it also means that the cover story may need to be updated—something that is not necessarily the case if only single level nodes are used.

The paper also assumed that updates to the system will only be done by specific persons in the organisation and that labelling occurs between composing the node and entering it into the system. However, it may be necessary to allow updates by users (or ‘comments’ in the words of one definition quoted in the introduction). Such updates will also have implications on the security system used—especially if information flow is to be restricted.

These problems remain interesting research questions.

REFERENCES

- Feldman, T (1994) *Multimedia*, Chapman & Hall, London
- Garvey, TD and Lunt, TF (1991) Cover Stories for Database Security, *Proceedings of the Fifth IFIP WG11.3 Workshop on Database Security*, Shepherdstown, USA
- Gay, G and Mazur, J (1991) Navigating in Hypermedia, in *Hypertext / Hypermedia Handbook*, (eds E Berk and J Devlin), McGraw-Hill, New York
- Lunt, TF (1990) Multilevel Security for Object-Oriented Database Systems, pp 199–209 in *Database Security III: Status and Prospects* (eds DL Spooner and C Landwehr), North-Holland, Amsterdam
- Maurer, H (1993) An Overview of Hypermedia and Multimedia Systems, pp1–12 in *Virtual Worlds and Multimedia* (eds NM Thalman and D Thalman), Wiley
- Kim, W (1991) Object-oriented Database Systems: Strengths and Weaknesses, *Journal of Object-oriented Programming*, **4**, 4, 21–29
- Olivier, MS and Von Solms, SH (1994) A Taxonomy for Secure Object-oriented Databases, *ACM Transactions on Database Systems*, **19**, 1, 3–46
- Olivier, MS (1994) A Multilevel Secure Federated Database, *Proceedings of the Eighth IFIP 11.3 Working Conference on Database Security*, Bad Salzdetfurth, Germany
- Olivier, MS (1995) Self-protecting Objects in a Secure Federated Database Submitted
- Pernul, G (1993) Canonical Security Modelling for Federated Databases, pp207–22 in *Interoperable Database Systems* (eds DK Hsiao, EJ Neuhold and R Sacks-Davis), Elsevier, Amsterdam
- Pfleeger, CP (1989) *Security in Computing*, Prentice-Hall, London
- Rabitti, F, Bertino, E, Kim, W and Woelk, D (1991) A Model of Authorization for Next-Generation Database Systems, *ACM Transactions on Database Systems*, **16**, 1, 88–131
- Sandhu, RS (1993) Lattice-based Access Control Models, *IEEE Computer*, 9–19
- Slatin, JM (1991) Composing Hypertext: A Discussion for Writing Teachers, pp55–64 in *Hypertext / Hypermedia Handbook* (eds E Berk and J Devlin), McGraw-Hill, New York
- Stein, RM (1994) Object Databases, *Byte*, 74–84
- Thuraisingham, B (1993) Multilevel Security for Information Retrieval Systems, *Information & Management*, **24**, 93–103
- Thuraisingham, B (1994) Security Issues for Federated Database Systems, *Computers & Security*, **13**, 509–25
- Thuraisingham, B (1995) Multilevel Security for Information Retrieval Systems – II, *Information & Management*, **28**, 49–61
- Van Dyke Parunak, H (1991) Ordering the Information Graph, pp299–325 in *Hypertext / Hypermedia Handbook* (eds E Berk and J Devlin), McGraw-Hill, New York