Examples of Lattices in Computer Security Models

[redacted] STATUTORILY EXEMPT

Formal models of secure computer systems use the algebraic concept of a lattice to describe certain components of the system. In this note three examples are presented: the space of security values in the Bell-LaPadula model, the space of multi-level objects, and Dorothy Denning's information flow model.

I. USE OF A LATTICE IN THE BELL-LAPADULA MODEL

In this note we point out several important notions in the area of computer security which constitute examples of the mathematical concept of a lattice. Perhaps the best known use of a lattice appears in the Bell-LaPadula model for a secure computer system as described in [1]. Let L denote the set of all security values that may be assigned to sensitive information to be handled within the system. It is important to be able to give precise meaning to the notion of one security value being superior to or "dominating" another security value. To accomplish this, Bell and LaPadula define L to be

$$L = C \times PK$$

where C is a finite simply ordered set and PK is the power set of another finite set K. The members of the set

$$C = C_1 > C_2 > \ldots > C_n$$

are the familiar designators C_1 = TOP SECRET; . . . ; C_n = UNCLASSIFIED. The set K consists of all the additional constraints which are placed on the dissemination of information such as codewords, special clearances, etc. Thus, a security value ℓ is of the form

$$\ell = C_i, k$$

where $1 \leqq i \leqq n$ and $k \subseteq K$. If $\ell_1 = C_{i_1}, k_1$ and $\ell_2 = C_{i_2}, k_2$, then ℓ_1 "dominates" ℓ_2 denoted by

$$\ell_1 \propto \ell_2$$

if and only if

$$i_1 \leqq i_2 \text{ and } k_1 \supseteq k_2$$

It is easy to see that the relation $\gg$ is reflexive, antisymmetric, and transitive; consequently, L is a partially ordered set. The least upper bound of ℓ_1 and ℓ_2 is

$$\text{lub}(\ell_1, \ell_2) = C_{i_3}, k_3$$

where $i_3 = \min(i_1, i_2)$ and $k_3 = k_1 \cup k_2$. Similarly, the greatest lower bound of ℓ_1 and ℓ_2 is

$$\text{glb}(\ell_1, \ell_2) = C_{i_4}, k_4$$

where $i_4 = \max(i_1, i_2)$ and $k_4 = k_1 \cap k_2$. The universal upper bound is (C_1, K) while the universal lower bound is $(C_n, 0)$. Hence, L is a lattice. For the case $n = 2$ and $K = 1, 2, 3$ the lattice L is described by the following diagram:

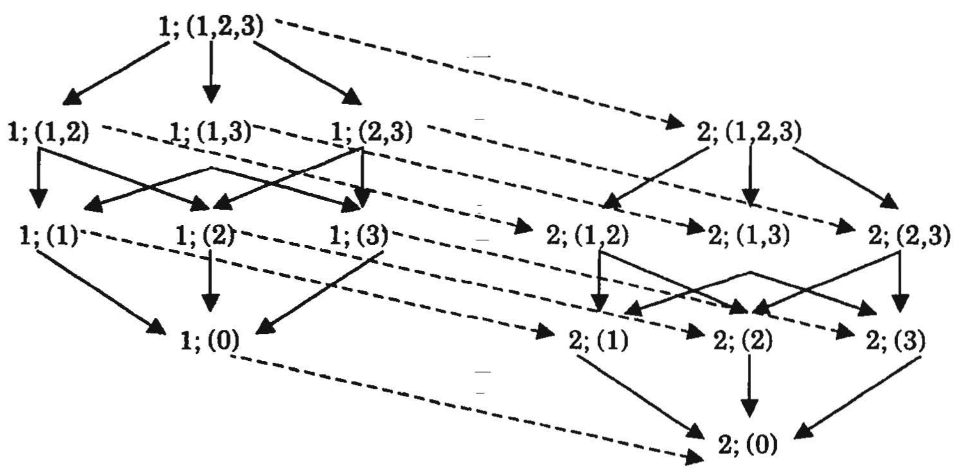

Fig. 1.

II. THE MULTI-LEVEL OBJECT MODEL AS A LATTICE

The units of information in the Bell-LaPadula model are termed objects and each object o is assigned a security value $f_o(o)$ which lies in L. A generalization is the case where objects are themselves collections of other information units whose security values differ, the multi-level object model. An example is a filing cabinet housing a collection of classified technical papers. The cabinet itself can be viewed as a collection of drawers; each drawer can be viewed as a collection of folders while each folder is a set of classified papers whose security values constitute a subset of L. The model for the family of objects is a hierarchy as defined in [2]. Formally, a hierarchy is a collection of sets

$$H = \{S_i;\, o \leqq i \leqq n\}$$

where $S_o \supset S_i$ for $1 \leqq i \leqq n$ and if $S_i \cap S_j \neq 0$, then either $S_i \supset S_j$ or $S_j \supset S_i$ for all $1 \leqq i \neq j \leqq n$. Clearly, the hierarchy H is a subset of PS_o, the power set of S_o. It is well known that PS_o is a lattice where the partial ordering relation is "contains"; hence H is a partially ordered set. If we annex the null set to H, the enlarged set H^* is a lattice. The diagram of H is a tree (no cycles) as is shown in the following example:

Suppose S_o is a set of 10 elements denoted by the integers 1, . . . ,10. Let the sets S_i be defined as follows:

$S_1 = (1,3,5)$	$S_4 = (1,5)$	$S_7 = (8,9,10)$
$S_2 = (2,4,6)$	$S_5 = (3)$	$S_8 = (8,9)$
$S_3 = (7,8,9,10)$	$S_6 = (2,6)$	$S_9 = (8)$
		$S_{10} = (9)$

The diagram representing H is

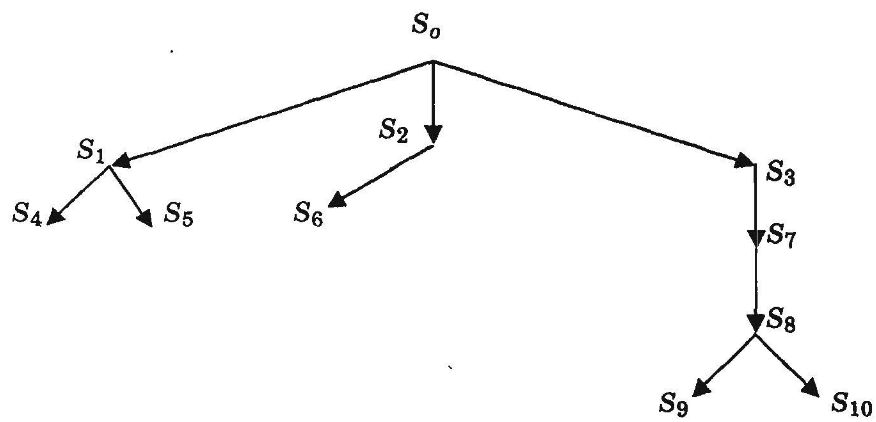

Fig. 2

The set of objects in the Bell-LaPadula model can be represented by a hierarchy: the set S_o is the set of all objects while each set S_i contains one and only one object. The diagram is the trivial tree

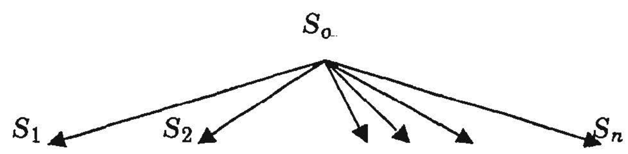

Fig. 3.

III. DENNING'S SECURE INFORMATION FLOW MODEL LATTICE

Possibly the most important use of lattices occurs in Dorothy Denning's model for secure information flow. By means of a set of processes P, information is said to "flow" among objects in the set N. For example, when a message is transmitted from terminal a to terminal b, information "flows" between a and b. When an object updates a file, information "flows" between the two. In general, when any object invokes any of the processes in P, an information flow is created. In order to describe security measures formally in this context, we first postulate that a function f has been defined which assigns to each object a in N a security value $f(a)$ in the set of security values SC.

Next, we postulate that the system designer describes the security policy of the system by means of a set of ordered pairs Ω in $SC \times SC$. The security policy is simply this:

> If $f(a) = A$ and $f(b) = B$, then information is allowed to flow from a to b if and only if the ordered pair (A,B) is in Ω.

This view of security seems to imply that, given the ordered pair (A,B), the security value B is in some way "superior" to A since it would violate all our intuitive feelings to allow information of security value A to flow to an object of lesser security classification. This notion is easily developed as will be shown below.

The set of ordered pairs $\Omega \subset SC \times SC$ is, of course, an example of a relation. There are several properties our intuition says Ω ought to possess if the relation is to provide a realistic description of "secure information flow." Certainly, there ought to be transitivity: if information can flow from A to B and also from B to C, then it ought to be safe to allow the flow from A to C. In the context of ordered pairs, this says that if (A,B) and (B,C) are in Ω, then so is (A,C). Also, if information can flow from A to B and B to A then A and B ought to be of equal weight or value. Consequently, we postulate antisymmetry. Further, it seems reasonable to postulate that if objects a_1 and a_2 both have security value A, then information can flow from a_1 to a_2. This, of course, says that the relation is reflexive. If we adopt the notation "$A \rightarrow B$" for (A,B), then, formally, we assume

(i) $A \rightarrow A$ for all $A \in SC$
(ii) $A \rightarrow B$ and $B \rightarrow A$ if and only if $A = B$
(iii) if $A \rightarrow B$ and $B \rightarrow C$, then $A \rightarrow C$

Thus, the set of security values SC, with respect to the relation "$\rightarrow$" is a partially ordered set. Without loss of generality we may assume that SC contains both a "greatest" member U and a "least" member L, i.e., for all $A \varepsilon SC$

$$L \rightarrow A \text{ and } A \rightarrow U.$$

L corresponds to the classification of information available to all entities in the universe; information classified U is available only to the ruling elite.

There is one additional problem the designer of a secure information flow system must solve. Suppose a process is able to generate a new information unit which is a function of the information contained in objects $a_1, a_2, \ldots, a_n$. The problem is to decide what security value to assign the new element we shall call

$$a^* = g(a_1, a_2, \ldots, a_n).$$

For example, a^* may be a report produced from data associated with $a_1, a_2, \ldots, a_n$ and may be upgraded periodically. Our intuition says that $f(a^*)$ ought not to be inferior to $f(a_i)$ for any $1 \leqq i \leqq n$; symbolically, we desire

$$f(a_i) \rightarrow f(a^*) \qquad\qquad 1 \leqq i \leqq n.$$

This implies that $f(a^*)$ is an upper bound for the set of elements

$$\{f(a_i) : 1 \leqq i \leqq n\}$$

in SC. In order to prevent overclassification, we would like to make $f(a^*)$ the "least" of the upper bounds. Consequently, we postulate that our set Ω satisfies the additional condition:

For every A and B in SC, there exists C in SC such that

(i) $A \rightarrow C$ and $B \rightarrow C$

(ii) if $D \varepsilon SC$ such that $A \rightarrow D$ and $B \rightarrow D$, then $C \rightarrow D$

All the conditions we have placed on the set of ordered pairs Ω enable us to prove that SC is a lattice with respect to the relation $\rightarrow$. (Recall that the greatest lower bound of A and B is the least upper bound of the set of all elements F such that $F \rightarrow A$ and $F \rightarrow B$.) Note the order in which this characterization has evolved. We began with the collection of ordered pairs Ω, specified by the system designer as the security policy. Reasonable constraints placed on Ω to satisfy our intuitive security concerns, when formalized, led to the mathematical characterization of SC as a lattice.

BIBLIOGRAPHY

[1] Bell, D. E. and L. J. LaPadula. "Secure Computer System: Unified Exposition and Multics Interpretation," Mitre Technical Report 2997, July 1975.

[2] Robinson, D. F. and L. R. Foulds. *Digraphs: Theory and Techniques.* New York: Gordon and Breach Science Publishers, 1980.

[3] Denning, D. E. "A Lattice Model for Secure Information Flow," *Communications of the ACM*, Vol. 19, No. 5, May 1976.