Analysis of Some Abstract Measures of Protection in Computer Systems*

Clarence A. Ellis

Department of Computer Science
University of Colorado
Boulder, Colorado 80302

Report #CU-CS-043-74 May 1974

* This work was submitted for publication to Acta Informatica. This work was supported by NSF Grant #GJ-660.

ABSTRACT

The area of computer systems protection has been acclaimed as a very important one, and there is a vast amount of literature on the subject (including several books). However, implementations of protection mechanisms tend to be ad-hoc and there is a lack of quantitative theoretical results upon which one can base decisions and abstract the essence of protection. This paper is the first to present a mathematically rigorous definition (with proofs) of the degree of protection of a system. It is hoped that this type of presentation will alert systems designers and implementors to alternatives to and generalizations of current implementations and to some of the trade-offs involved in these alternatives. Ultimately, it is hoped that this presentation will contribute to a general Theory of Protection.

The investigation is directed toward an analysis and comparison of access mechanisms defined by a family of boolean functions. Some definitions are stated, and some theorems are proved which are valid for all access mechanisms within the family considered. Algorithms are presented for the optimal assignment of access codes to subjects and objects for unstructured systems and for several types of structured systems. It is proven that for a very general class of systems, the optimal assignment will still allow $\frac{n}{2}(\gamma-1)$ unauthorized accesses to objects where n is the number of subjects and γ is the largest integer not greater than the quantity n divided by the number of access classes.

Key words. protection, access mechanisms, security, sharing, hierarchical systems, privacy, access control, storage protection keys, passwords

1. INTRODUCTION

For any given degree of protection within a computer system, there is a cost associated with that protection in terms of dollars to build and maintain the system and in terms of possible loss of capability, [2, 19]. As an example, consider a file system which does not support protection on a sub-file level. Thus a user will have access to all of a file or none of it. For large file systems, the cost of an ideally secure system having access protection down to the bit level [8] may not be cost effective. Thus users having a need to access various portions of a file would all be given access to the entire file. This is a case of placing many subjects (the users) into a single access class. This could be quite bad if, for example, the access was write access to an employee file, and a program to increase the weekly pay rates of one class of employees inadvertently zeroed the pay rates of another class of employees. Another example encountered by the author concerns co-ordinate verification data received from radar by an air traffic control computer. It was frequently found that the least significant k bits were not exact. Thus the system ignored these bits since they could be re-created elsewhere in the system with reasonable accuracy if needed. This lumping together of many co-ordinate points is again an example of placing many subjects (the co-ordinate points) into fewer access classes. This example generalizes to arbitrary data transmission of identifiers or passwords such that a bit could be lost. If 1 bit (k bits in general) error detection is not employed, then two passwords differing by 1 bit (k bits) or less are being placed into the same access class. The final example concerns the assignment of numbers to automatic answer back drums of remote terminals. The case where several terminals have the same number means that user numbers validated for a particular terminal may not be blocked from entering the system from the wrong terminal. These are all examples of the principle put forth by Peters [16] stating that "security cannot be attained in the absolute sense. Every security system seeks to attain a probability of loss which is commensurate with the value returned by the operation being secured".

Thus conceeding that there are many computer applications (and non-computer applications) where a perfectionist approach of absolute protection is untenable, this paper attempts to quantify the notion of degree of protection of a system in terms of numbers of unauthorized accesses due to access classes containing more than one subject.

Terminology and definitions used throughout the paper are given in the next section. Each access mechanism in the family under consideration will allow access by a subject to an object contingent upon a calculation which is made upon the "access codes" of the subject and object in question. This class of mechanisms will be defined, illustrated, and its characteristics elucidated in section 2. This family basically consists of boolean functions applied bit-wise to access codes and summed. The sum is then compared to a threshold to determine if access is to be granted. It is shown that particular cases within the family yield mechanisms such as the very common password scheme which are used in some well-known systems, [9,6] It is hoped that this type of presentation will alert hardware designers and systems implementers to alternatives and possibilities of generalizing current implementations and to some of the trade-offs involved in doing so. Given that one of these access mechanisms has been selected, it is then necessary to assign access codes to subjects and objects of the system. In section 3, systems such that all subjects are isolated are studied. An assignment algorithm is presented and a proof is given that this algorithm maximizes the (absolute, relative, and minimum) degree of protection of the system. The assumption of isolation implies that no authorized sharing of data or hierarchy of processes exists within the system. In section 4, this assumption is relaxed and some bounds are derived for systems containing arbitrary partial and/or total hierarchies. Assignment algorithms are given for ring structures and for binary tree structures. The conclusion is reached in the summary that choice of both the access mechanism and the assignment algorithm may have significant bearing upon the efficiency and security of a system. Finally, some possible directions of future research are indicated.

2. DEFINITIONS AND TERMINOLOGY

In this, and the following sections, we are not concerned with security in its most general form. The scope of this field encompasses so many factors [15, 11], so many of which remain unquantified [7, 18] that a narrower approach is warranted. Thus, this paper simply considers a systems design problem whose solutions within various subsystems have ramifications for the total security of the system. We are not directly concerned with protection procedures and policies except that within these procedures it may be deemed necessary to make validity checks before allowing certain communications or accesses. Also, policies will dictate which accesses are authorized. The following pages address themselves to the nature of these checks in cases where an automated password type of mechanism is applicable. Results presented are general in that they hold for a large class of systems and access mechanisms. They could be called uninterpreted in the sense that we leave unspecified the semantics of the environment in which an access mechanism is being used and assume this environment is free from external tampering. Questions of overall system policies, certification of subsystems, changes of environment, and the like, are semantic questions not investigated in this paper. This paper defines and investigates a purely syntactic quantity called the <u>degree of protection</u> of an (uninterpreted) system.

The pertinent elements of a system (A,B,g) for our characterization of security are the following. Denote a finite set of active elements, called <u>subjects</u> [4] by $A = \{A_1, A_2, \ldots, A_u\}$ and a finite set of passive elements, called objects, by $B = \{B_1, B_2, \ldots, B_v\}$. Certain subjects may be specified to have relationships, called authorized accesses, to certain objects. It may occur that an element is both active and passive, so in general, $A \cap B \neq \emptyset$ and we will denote $C = A \cup B$. This formalism can also be used to model a network of communicating processors by setting A=B and interpreting access as communication between subjects. Associated with each element $C_i \in C$ is an <u>access code</u> c_i which is a binary n-digit number. We will call an access code a_i for a subject A_i and b_i for an object B_i. A set of subjects all of which have the same access code will be defined as an access class. The k-th binary digit of an access code c_i will be denoted c_{ik}. An access mechanism is a total boolean function of an ordered set of two access codes, $g(a_i,b_j)$, which takes on a value of 1 if access from A_i to B_j is allowed and zero if access is not allowed. Note that the first code a_i must correspond to an active element, and the second, b_j, to a passive element. The following family of access mechanisms will be studied in this paper.

$$(2.1) \quad g = 1 \quad \text{if} \quad \sum_{k=1}^{n} f(a_{ik}, b_{jk}) \geq m$$

$$g = 0 \quad \text{otherwise}$$

where f is an arbitrary Boolean function of two binary digits,

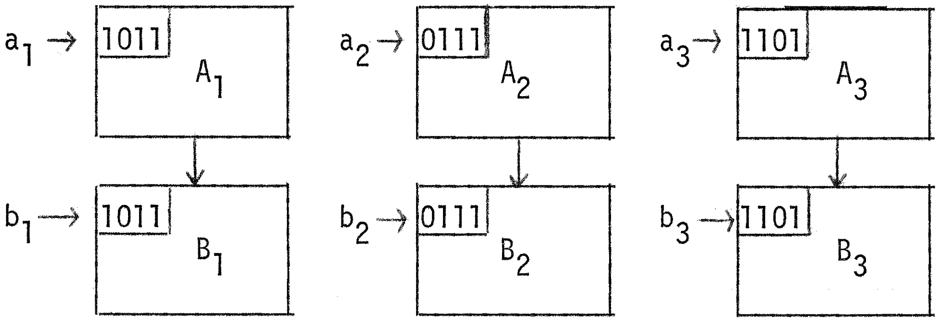
n is the number of bits in an access code, and

m, called the access threshold, satisfies $0 \leq m \leq n$.

An example is in order:

Example 1. Suppose a computer system has a memory protection mechanism consisting of hardware keys and locks. Whenever a process attempts to access a memory, an automatic check is made and access is granted if and only if the process key matches the memory lock. Assuming 4 bit registers for locks and keys, the diagram below (figure 1) shows a system of three processes and three memories such that process A_i has exclusive access to memory B_i, i = 1,2,3.

Figure 1.



This scheme of exact match, which is employed within the IBM 360/370 system [9] and other systems for memory protection, is one particular instance of the formula (2.1) obtained by choosing $f(u,v) = (u \equiv v)$ and m=n=4.

$$(2.2) \quad g = 1 \quad \text{if} \quad \sum_{k=1}^{4} a_{ik} \equiv b_{jk} \geq 4$$

$$g = 0 \quad \text{otherwise.}$$

The exact match mechanism, which yields the equation of type (2.2), is also the basis for password systems. The maximum number of subjects supported such that each can have exclusive access to one or more objects is 16. A set of subjects such that no sharing of objects or hierarchy of subjects is desired and with $A \cap B = \phi$ will be called isolated subjects. Similarly, a set of access codes will be called isolated codes with respect to a particular access mechanism if assignment of all of these codes to subjects implied that there exists an assignment of codes to objects such that each subject has exclusive access to one or more objects. Denote the maximum number of isolated n bit access codes obtainable by a given access mechanism f with threshold m by $S_n^m(f)$, called the isolation level of the mechanism. The access mechanism of example 1 has an isolation level of $S_4^4(\equiv) = 16$. Generalizing m = 4 in a system with four bit access codes to an exact match scheme for an arbitrary n bit access code yields an isolation level equal to the number of distinct n bit codes, $S_n^n(\equiv) = 2^n$. An access mechanism can also be chosen to support the sharing of memory (or arbitrary resources) between two or more processes. Suppose we choose f(u,v)=(u NOR v) and m = 1, yielding:

$$(2.3) \quad g = 1 \quad \text{if} \quad \sum_{k=1}^{n} (a_{ik} \text{ NOR } b_{jk}) \geq 1$$

$$g = 0 \quad \text{otherwise.}$$

Then resources with an access code of 0011 could be automatically shared by processes A_1 and A_2 of figure 1 while not allowing access by A_3. Similarly, any other sharing arrangement can be implemented. Hierarchies of processes can also be implemented with this scheme; for example a supervisory process requiring access to all resources of figure 1 could have a master key of 0000. This mechanism, one form of which is employed within the RC 4000 system [6], allows automatic sharing of objects and arbitrary hierarchies of subjects which was not possible with the exact match mechanism. However, in example 1 the isolation level of the system is reduced from 16 to 4; for arbitrary length access codes the reduction is from $S_n^n(\equiv)=2^n$ to $S_n^1(\overline{v})= n$. This is tremendous, and a compromise suggested by our generalization (2.1) which allows sharing and hierarchies is to choose a NOR mechanism with $m > 1$. This allows a larger number of isolated access codes, namely $S_n^m(\overline{v})=\binom{n}{m}$ = the number of combinations of n items taken m at a time, which is maximized at $m = n/2$. We note in passing that there are also some access mechanisms within our family that would be rather poor design choices. For example if function f is the less than (<) function and $m = n$, then there is no isolation capability (i.e. $S_n^m(f) = 1$). The trivial functions $f(u,v) = 0, = 1, = u,$ and $= v$ all have no isolation capability and will be disregarded throughout this paper.

The systems of primary interest in this paper are ones in which the maximum number of isolated subjects is greater than S_n^m for the access mechanism chosen for the system. In this case one or more subjects may be assigned access codes which allow access to objects to which the subjects should not have access. We define this type of access as an unauthorized access. If the system is one requiring sharing or hierarchies of subjects, then this fact may be used advantageously to decrease or even eliminate unauthorized accesses. Thus if two subjects require shared use of one or more objects and do not require isolated access to any objects, then they could be placed in the same access class, i.e. assigned the same access code. Any access ability, such as this, of a subject to an object which is not an unauthorized access is termed an authorized access. Given a set of subjects, a set of objects, their hierarchy constraints, and an access mechanism within our family, our central problem is the determination of how best to assign access codes to subjects and objects so that all required authorized accesses are allowed and a minimum number of unauthorized accesses are allowed. To this end, we specify the following definitions:

1. Let x_{ij} and y_{ij} be boolean variables such that $x_{ij} = 1$ ($y_{ij} = 1$) implies that A_i has authorized (unauthorized) access to B_j; $x_{ij} = 0(y_{ij} = 0)$ otherwise. $x_{ij} \wedge y_{ij} = 0$ necessarily for all $i(1 \leq i \leq |A|)$ and for all $j(1 \leq j \leq |B|)$ where $|Z|$ denotes the cardinality of the set Z.
2. Let $x_j(y_j)$ be the number of $A_i \in A$ which have authorized (unauthorized) access to $B_j \in B$, i.e.

   a) $x_j = \sum_{A_i \in A} x_{ij}$, b) $y_j = \sum_{A_i \in A} y_{ij}$

3. Let $\overline{x}$ ($\overline{y}$) be the average, over B, of the number of subjects which have authorized (unauthorized) access to any particular $B_j \in B$, i.e.

   a) $\overline{x} = \dfrac{\sum_{B_j \varepsilon B} x_j}{|B|}$, b) $\overline{y} = \dfrac{\sum_{B_j \varepsilon B} y_j}{|B|}$.

4. Let $\check{x}$ ($\check{y}$) be the minimum, over B, of the number of subjects which have authorized (unauthorized) access to any particular $B_j \in B$, i.e.

   $\check{x} = \min_{B_j \in B} (x_j)$, $\check{y} = \min_{B_j \in B} (y_j)$

5. Let $\hat{x}$ ($\hat{y}$) be the maximum, over B, of the number of subjects which have authorized (unauthorized) access to any particular $B_j \in B$, i.e.

   a) $\hat{x} = \max_{B_j \in B} (x_j)$, b) $\hat{y} = \max_{B_j \in B} (y_j)$

6. The absolute degree of protection of a system is then

   $\delta_{abs} = (1 + \overline{y})^{-1}$

This definition yields a value of unity when the system allows no unauthorized accesses, and less than unity otherwise. It is absolute in the sense that it does not vary according to the size of the system, but only according to the average number of unauthorized accesses.

7. The relative degree of protection of a system is

$$\delta_{rel} = \frac{|A| - \overline{x} - \overline{y}}{|A| - \overline{x}}$$

This definition yields a value of unity when the system allows no unauthorized access, and a value of zero when the maximum possible number of unauthorized accesses are allowed. This value is therefore relative to the size and structure of the system being analyzed.

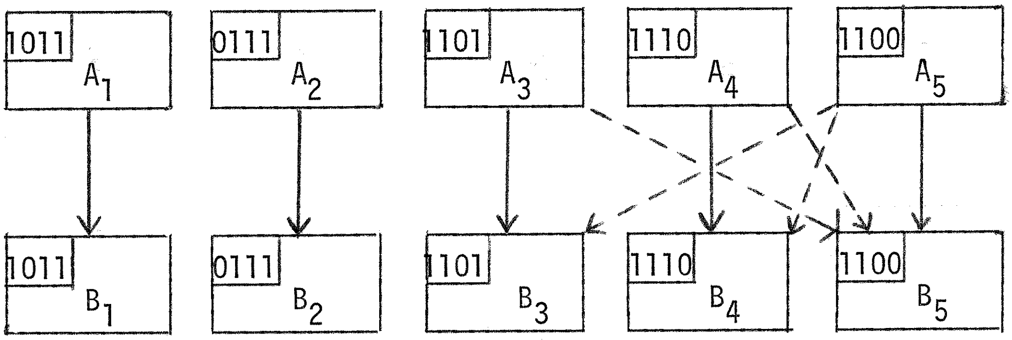
8. The minimum (maximum) degree of protection of a system, $\check{\delta}$ $(\hat{\delta})$ is

a) $\check{\delta} = (1 + \hat{y})^{-1}$

b) $\hat{\delta} = (1 + \check{y})^{-1}$

Example 2. Consider a system of five isolated subjects ($|A| = 5$) and five objects ($|B| = 5$) such that each subject A_i requires exclusive access to exactly one object B_i. Assume that the access mechanism is specified by a NOR function with $n = 4$, $m = 1$. As previously stated the maximum number of isolated subjects which can be supported such that each can have exclusive access to one object is $S_4^1(\overline{V}) = 4$. Thus any assignment of access codes to the five subjects and objects implies some unauthorized access under the given mechanism. Figure 2 shows one possible assignment of access codes. Is this the best possible?

Figure 2.



In figure 2, authorized accesses are denoted by solid directed lines, and unauthorized accesses by dotted lines. In terms of the previous definitions, the following values are obtained for this system with the given access code assignment.

1. $x_{ij} = \delta_{ij}, 1 \leq i \leq 5, 1 \leq j \leq 5$, (where δ_{ij} is the Kronecker Delta meaning

   $x_{ij} = 1$ if $i = j$, $x_{ij} = 0$ otherwise).

   $y_{35} = y_{45} = y_{53} = y_{54} = 1$, all other $y_{ij} = 0$.

2. $x_j = \sum_{i=1}^{5} x_{ij} = x_{jj} = 1, 1 \leq j \leq 5$

   $$y_j = \sum_{i=1}^{5} y_{ij} \Rightarrow \begin{aligned} y_1 &= 0 \\ y_2 &= 0 \\ y_3 &= y_{53} = 1 \\ y_4 &= y_{54} = 1 \\ y_5 &= y_{35} + y_{45} = 2 \end{aligned}$$

3. $\bar{x} = \sum_{j=1}^{5} x_{j/5} = 5/5 = 1$

   $\bar{y} = \sum_{j=1}^{5} y_{j/5} = \frac{1 + 1 + 2}{5} = 4/5$

4. $\check{x} = \min_{B_j \varepsilon B} (x_j) = 1$

   $\check{y} = \min_{B_j \varepsilon B} (y_j) = 0$

5. $\hat{x} = \max_{B_j \varepsilon B} (x_j) = 1$

   $\hat{y} = \max_{B_j \varepsilon B} (y_j) = 2$

6. $\delta_{abs} = (1 + \bar{y})^{-1} = 5/9$

7. $\delta_{rel} = \frac{5 - \bar{x} - \bar{y}}{5 - \bar{x}} = 4/5$

8. $\check{\delta} = (1 + \hat{y})^{-1} = 1/3$

   $\hat{\delta} = (1 + \check{y})^{-1} = 1$

This is not the best possible assignment of access codes to subjects and objects for this example. By making a change of access code assignments to subject A_5 and B_5so that $a_5 = b_5 = 1110$, an optimal system is obtained in that it minimizes the number of unauthorized accesses and maximizes δ_{abs}, δ_{rel}, and $\check{\delta}$ as follows:

$$\bar{\delta}_{abs} = 5/7$$

$$\bar{\delta}_{rel} = 9/10$$

$$\check{\delta} = 1/2.$$

Note that in this case, with this access mechanism, the optimal access code assignment was obtained by always choosing codes which have a minimum number of bits set to allow access (in this case a set bit means a zero bit). This will be proved for all cases and all access mechanisms within our family by proof la of the next section. Furthermore, an optimal assignment must distribute subjects (and their objects) as evenly as possible throughout the available access classes. In this example we obviously shouldn't put more than two subjects in an access class. This notion of even distribution will be proved in general by proof lb of the next section.

3. UNSTRUCTURED SYSTEMS CONSIDERATIONS

Theorem 1: Given a system (A,B,g) consisting of

(a) a set of isolated subjects,

(b) a set of objects such that $|A| = |B|$, and

(c) a monotonic access mechanism of the form:

A_i has access to B_j iff $\sum_{k=1}^{n} f(a_{ik},b_{jk}) \geq m$ with $S_n^m(f) < |A|$.

Then an access code assignment satisfying $x_j = 1$ for all $B_j \varepsilon B$ maximizes the absolute and relative degrees of protection of the given system

iff

(a) the assignment uses $S_n^m(f)$ isolated minimal access code pairs, and

(b) the assignment distributes subjects as evenly as possible into access classes.

The proof of these two conditions will be presented in two parts as Proof 1a and Proof 1b. First some terminology needed within the proof must be explained.

A non-decreasing access mechanism is one such that $u' \geq u$ and $v' \geq v$ implies $f(u',v') \geq f(u,v)$.

A non-increasing access mechanism is one such that $u' \geq u$ and $v' \geq v$ implies $f(u',v') \leq f(u,v)$.

A monotonic access mechanism is one which is either non-decreasing or non-increasing.

Reference will be made to a bit of an access code being set. For a non-decreasing mechanism this means the bit equals 1; for a non-increasing mechanism this means the bit equals 0. In both cases reset means the opposite value from set. Define the field length of an assignment as $|F|$ where $F = \{k| \ 1 \leq k \leq n$ and

$\exists a_i, b_i \ni f(a_{ik},b_{ik}) > 0\}$.

Define the minimum field length for a system (A,B,g) to be the minimum of $|F|$ over all assignments which have $S_n^m(f)$ isolated access codes. For the function Λ, all bits are needed to get the maximum number of isolated codes, so that minimum field length = n; for the function V, only m bits are needed so the minimum field length

= m. In general, the minimum field length depends upon the particular function and therefore its study is relegated to an appendix. Appendix A shows that we never need consider code words of length greater than the minimum field length, so this assumption is made throughout the remainder of the paper unless another assumption is explicitly stated. Define α_i and β_i as the number of bits set in the access codes a_i and b_i respectively. Let α_{min} denote the minimum min (α_i) where this minimum is taken over all access codes of all assignments which have exactly $S_n^m(f)$ isolated codes. Let β_{min} denote the minimum number of bits which must be set in b_i for it to be accessed by a_i assuming a_i has α_{min} bits set. Some characteristics of α_{min} and β_{min} are next mentioned:

a. For the $\wedge$ function $\beta_{min} = \alpha_{min}$; for the V function, $\beta_{min_n} = m - \alpha_{min}$.

b. If a_i has α_{min} bits set and b_i has β_{min} bits set and A_i has access to B_i, then the function value must be the minimum possible which still allows access, so

$$\sum_{k=1}^{n} f(a_{ik}, b_{ik}) = m$$

c. Since we are assuming a number of bits equal to the minimum field length, for any a_i having exactly α_{min} bits set, there exists one and only one access code having β_{min} bits set such that $\sum_{k=1}^{n} f(a_{ik}, b_{jk}) = m$.

d. Similarly for any b_i satisfying above constraints, there is a unique a_i determined by b_i. Call an a_i, b_i satisfying these constraints a <u>minimal code pair</u>.

e. <u>Isolated minimal access code pairs</u> are defined as minimal code pairs such that all a_i are isolated.

It is possible to make assignments to $S_n^m(f)$ isolated subjects and their objects in such a way that there are no unauthorized accesses by only using isolated minimal access code pairs. This implies the degree of protection is one ($\delta_{abs} = \delta_{rel} = 1$). But when the number of isolated subjects is greater than S_n^m (f), it is not obvious, in general, whether it is useful to introduce other access codes containing more than $\propto_{min}$ (or β_{min}) set bits. The following proof shows that assignments containing more than the minimal number of set bits are always sub-optimal because δ_{abs} and δ_{rel} can always be increased.

Proof 1a:

It will be shown that any assignment which maximizes the absolute and relative degrees of protection of a system (of $|A| > S_n^m$ (f) isolated subjects) contains only isolated minimal access code pairs by hypothesizing that some given assignment is optimal and contains at least one access code such that $\beta_i > \beta_{min}$. A similar argument applies for the assumption $\propto_i > \propto_{min}$. In each case, an inconsistency is obtained yielding a proof by contradiction.

step 1: By assumption, $\exists$ an object, call it $B_o \ni \beta_o \geq \beta_{min} + 1$ within the given hypothetically optimal assignment. Name as A_o the subject which has authorized access to B_o. Similarly name all A_i and B_i so that $x_{ii} = 1$, $1 \leq i \leq n$.

step 2: By definition of β_{min}, $\exists$ at least one bit (call it bit r) and in general $\beta_o - \beta_{min}$ bits of b_o which can be reset and still have access by a_o. Reset these bits and call the altered access code b_o' . No new unauthorized accesses were introduced by this alteration because bits were only reset which cannot increase $f(a_{ik}, b_{ok})$ by the monotonicity of the access mechanism.

step 3: If $\propto_o > \propto_{min}$, reset the excess bits to form the unique minimal a_o' corresponding to b_o'. Since $\Sigma f(a_{ok}, b_{ok}') \geq m$ it is guaranteed that no new bits need be set to form the a_o' such that $\Sigma f(a_{ok}', b_{ok}') = m$. Thus it can be asserted that $\Sigma f(a_{ok}, b_{ik}) \geq \Sigma f(a_{ok}', b_{ik}) \forall$ indices i. This means no new unauthorized accesses occurred which implies y_{ij} not increased which implies $\overline{y}$ not increased.

step 4: Suppose $\exists a_j \ni \Sigma f(a_{jk}, b_{ok}) \geq m$ but $\Sigma f(a_{jk}, b_{ok}') < m$. This decrease in unauthorized accesses implies a decrease of y_{jo} which implies a decrease of y_o

which implies a decrease of $\bar{y}$ since step 3 allowed no offsetting increase of unauthorized accesses. Since $\bar{\delta}_{abs}$ and $\bar{\delta}_{rel}$ are monotonic decreasing functions of $\bar{y}$, an increase in these functions is implied. Thus, by the hypothesis that these functions were already maximized, this case cannot occur.

step 5: Suppose $\exists\ a_j \ni \Sigma f(a_{jk}, b'_{ok}) \geq m$ but $a_j \neq a'_o$. By the uniqueness of minimal code pairs, $\alpha_j > \alpha_{min}$. Attempt to reset any bit of a_j which is set in a'_o, and still retain access to b_j. If this is possible, then doing so causes a reduction of the number of illegal accesses to b'_o, thus reducing y_{jo}. This implies a reduction of $\bar{y}$ and a corresponding increase in δ_{abs} and δ_{rel}. Thus, by hypothesis, this case cannot occur.

step 6: Reset bits of a_j until obtaining $a'_j \ni \alpha'_j = \alpha_{min}$. Choose bits in such a way that $\Sigma f(a'_{jk}, b_{jk}) \geq m$. By step 5 above, none of these reset bits were set in a'_o. By the uniqueness of minimal code pairs, $a'_j = a'_o$. Thus it is possible to reset zero or more bits of b_j to obtain $b'_j = b'_o$.

step 7: Repeat step 6 until all $a_j \ni \Sigma f(a_j b'_o) \geq m$ have been converted to $= a'_j = a'_o$ and $b'_j = b'_o$. In all of these steps no new bits are set implying no new unauthorized accesses implying no increase in $\bar{y}$ implying no decrease in the degree of protection.

step 8: Reduce all remaining b_i to b'_i such that $\beta'_i = \beta_{min}$ by resetting bits. Again, this causes no decrease in the degree of protection. If this step also causes no increase in the degree of protection (as implied by our hypothesis), then the following step will cause an increase.

step 9: Choose any $a_z, b_z \ni y_z > 0$. Since $|A| > S^m_n(f)$, at least one such pair exists. Fabricate a new access code for B_z by setting all but one of the bits of b_z which correspond to the bits set in b'_o. Also set bit r (defined in step 2) and reset all other bits of b_z to form b'_z. Since b'_z has exactly β_{min} bits set, $\exists$ a unique access code which forms a minimal code pair with b'_z. Assign this access code to the subject A_z and call the code a'_z. Since a'_z is minimal (in terms of bits set), and all b_i are minimal, A_z can only access objects with $b_i = b'_z$. If there were any code which could have been reduced to b_z', it would have been handled by the bit resetting of step 5, so no unauthorized accesses can be made by A_z. Furthermore there are no subjects which have unauthorized access to B_z because steps 4 and (5,6,7) respectively dispose of (a) cases of access to object B_z under code b_z' but not B_o under code b_o' and (b) cases of access to both B_z and B_o. Thus y_z is decreased by this step to $y_z = 0$ implying an increase of δ_{abs} and δ_{rel}. This is a contradiction of the hypothesis, so an optimal assignment $\Rightarrow$ the assignment uses only minimal access code pairs.

Proof 1b:

This proof shows that an assignment is optimal iff it is an assignment of the maximum number of minimal access code pairs which distributes subjects as evenly as possible into these S_n^m (f) access classes. The technique used is to formulate the problem as a nonlinear optimization problem and solve it using partial derivative calculations.

First it is noted that the problem can be viewed as one of placing subjects into access classes in such a way as to minimize the number of unauthorized accesses. Thus one obviously wants as many classes as possible. The number of access classes $S_n^m(f)$ will be abbreviated to S. Denote the number of subjects in the k-th class by γ_k. Our constraint equation is then $|A| = \sum_{k=1}^{S} \gamma_k$. The number of unauthorized accesses within any class k is $\gamma_k(\gamma_k-1)$ because each object in the class has γ_k-1 unauthorized accessors and one authorized accessor, and there are γ_k objects (whose subjects are) in the class. Summing to get the total number of unauthorized accesses yields $h(\gamma_1 \ldots \gamma_S) = \sum_{k=1}^{S} \gamma_k(\gamma_k-1)$ as the objective function to be minimized. This is equivalent to minimizing $\frac{\sum_{k=1}^{S} \gamma_k(\gamma_k-1)}{|A|} = \bar{y}$, so this minimization maximizes δ_{abs} and δ_{rel}.

Objective Function: $h(\gamma_1, \gamma_2 \dots, \gamma_S) = \sum_{k=1}^{S} \gamma_k(\gamma_k-1)$

Constraint: $|A| = \sum_{k=1}^{S} \gamma_k$

This function must have $\frac{\partial h}{\partial \gamma_i} = 0$ at any extremal point for i = 1,2,...,S.

Thus we substitute and take partial derivatives:

$$\gamma_1 = h_2(\gamma_2, \gamma_3 \dots \gamma_s) = |A| - \sum_{k=2}^{S} \gamma_k$$

$$0 = \frac{\partial h}{\partial \gamma_i} = \frac{\partial}{\partial \gamma_i}(h_2(h_2-1)) + \frac{\partial}{\partial \gamma_i}\Big(\sum_{k=2}^{S} \gamma_k(\gamma_k-1)\Big),\ 1 < i \leq S.$$

$$= (2h_2-1)\frac{\hat{\partial} h_2}{\partial \gamma_i} + \frac{\partial}{\partial \gamma_i}(\gamma_i(\gamma_i-1))$$

$$= -2|A| + 2\sum_{k=2}^{S} \gamma_k + 2\gamma_i$$

This implies $\gamma_i = |A| - \sum_{k=2}^{S} \gamma_k = \gamma_1$, j = 2,3,...,S,

so our solution is $\gamma_1 = \gamma_2 = \dots = \gamma_S = 1/|A|$.

This solution is unique and we need only argue that the extremal value of h at our solution point is a minimum. All second partial derivatives take on a value of $2\delta_{ij}$, $\frac{\partial^2 h}{\partial \gamma_i^2} = 2$, $\frac{\partial^2 h}{\partial \gamma_i \partial \gamma_j} = 0 (i \neq j)$. Thus, the Hessian matrix of second partial derivatives is positive non zero only on the diagonal implying a positive value for its determinant. This indicates the presence of a minimum at the solution point.

If the quantities S and |A| are such that it is impossible to assign exactly the same number of subjects to each access class (|A| doesn't evenly divide S), then all classes should contain [S/|A|] subjects except for (remainder S/|A|) classes which should contain [S/|A|] + 1 subjects. This is the case in which the optimal solution of h is non-integer and we must seek integer solutions which are a minimum distance from the optimal non-integer solution. We can justify that these solutions, as described above, are optimal by the following argument. Any distribution of subjects among access classes can be obtained from one of the almost even distributions which we claim to be optimal by a sequence of transformations. Each transformation moves a subject out of a class e_1 containing γ_1 subjects into a class e_2 containing γ_2 subjects with $\gamma_1 \leq \gamma_2$. After the transformation, e_1 contains γ_1-1 subjects and e_2 contains γ_2 + 1 subjects. The fact that each transformation disperses a pair of classes implies that the net effect of a transformation is to increase the number of unauthorized accesses. The number of unauthorized accesses before a transformation is:

$[\gamma_1(\gamma_1-1)] + [(\gamma_1+k)(\gamma_1+k-1)] = 2\gamma_1(\gamma_1+k-1) + k(k-1)$ where $k \geq 0$.

The number after the transformation is:

$[(\gamma_1-1)(\gamma_1-2)] + [(\gamma_1+k+1)(\gamma_1+k)] = 2\gamma_1(\gamma_1+k-1) + k(k-1) + 2(k+1)$.

Thus the number of unauthorized accesses increases by 2(k+1) during each transformation verifying that the proposed integer solutions are optimal.

Theorem 2: Given a system (A,B,g) consisting of

(a) a set A of isolated subjects,

(b) a set B of objects such that $|B| = |A| = u$, and

(c) a monotonic access mechanism within the family under consideration with $S_n^m(f) < |A|$.

Then any minimum unbiased assignment as detailed in the statement of theorem 1 maximizes the minimum degree of protection $\check{\delta}$. The quantity $\check{\delta}$ gives a worst case measure of the amount of protection of a system. A user can be sure that he will receive no worse treatment than that dictated by $\check{\delta}$ (and possibly much better treatment if $\check{\delta}$ is much larger than $\check{\delta}$. A well-designed system should maximize $\check{\delta}$, but $\check{\delta}$ can be quite high without indicating that the degree of protection, in general, of the system is high, e.g. one object of many could have no unauthorized accessors but all other objects may be indiscriminately accessed by any subject. This system would still have $\hat{\delta} = 1$ although $\check{\delta}$ and $\overline{\delta}$, which are better indicators, would be quite low. Thus we do not insist upon maximation of $\hat{\delta}$. This quantity is useful, however, in the dynamic assignment of access codes to new subjects entering a system. An access class obtaining the minimum $\check{y}$ corresponding to the maximum degree of protection $\hat{\delta}$ is an excellent candidate for adding the new subject and its object. Also $\hat{\delta} = 1$ is a reasonable requirement to impose within an interpreted system so that a very precious object B_j will be sure to have $y_{ij} = 0$ for all undesirable i.

Proof 2:

The proof is by induction on n, the number of bits in an access code. Building upon the previous proof, we need only consider assignments having some

$a_i > \alpha_{min}$ or $\beta_i > \beta_{min}$.

Case 1: $n = 1$

Since it is required that every subject have authorized access to one object, either $m = 0$ or at least one bit which is the only bit must be set in each access code c_i. Our induction hypothesis is that $\hat{y} = \left\langle \frac{u}{S_n^m} \right\rangle$ where $\langle z \rangle$ denotes the largest integer less than z. More generally, if there are u subjects, and $k(<u)$ minimal code pairs then $\hat{y} = \left\langle \frac{u}{k} \right\rangle$. Since S_1^1 and S_1^0 are both equal to one, $\left\langle \frac{u}{S_n^m} \right\rangle = \langle u \rangle = u-1$. Thus, the induction hypothesis is true for n=1 because for all possible values of m(m=0 or 1), all subjects have access to all objects, but authorized access to only one. The number of unauthorized accessors to each object is u-1, and the maximum, $\hat{y}$, is therefore u-1.

Case 2: $n > 1$

Assume the induction hypothesis is true for all $n' < n$ and further suppose that there is an assignment which is better, i.e. $\hat{y} * < y = \left\langle \frac{u}{S_n^m} \right\rangle$: We simply show that existence of $\hat{y}^*$ violates the induction hypothesis.

Since any minimum unbiased assignment yields $\hat{y} = \left\langle \frac{u}{S_n^m} \right\rangle$, the assignment yielding $\hat{y}^*$ must not be minimal so there is some object B_i with $\beta_i > \beta_{min}$ or some subject with $\alpha_i > \alpha_{min}$. We assume the former; an analogous argument can be given for the latter. Eliminate A_i, B_i and all pairs $A_j, B_j \ni \sum_{k=1}^{n} f(a_{jk} b_{ik}) \geq m$ from the system

By properties of $\hat{y}^*$, the number of subjects eliminated is $\leq \left\langle \frac{u}{S_n^m} \right\rangle$. The number of subjects remaining is $\geq u - \left\langle \frac{u}{S_n^m} \right\rangle$. Also, at least $\beta_{min} + 1$ bits were set in b_i implies some number $T_n^m \geq 1$ of minimal patterns cannot be used by remaining subjects and objects (assuming $m \neq o$). For this new system, the induction hypothesis is applicable yielding

$$\hat{y}^* \geq \left\langle \frac{u - \left\langle \frac{u}{S_n^m} \right\rangle}{S_n^m - T_n^m} \right\rangle$$

We show that this quantity is greater than or equal to $\left\langle \frac{u}{S_n^m} \right\rangle$ implying the desired contradiction. Note that if the denominator $S_n^m - T_n^m$ equals zero, then the number of subjects having access to B_i is u (all subjects). Thus $\hat{y}^* = u-1 \geq \left\langle \frac{u}{S_n^m} \right\rangle$, implying the desired contradiction. If $S_n^m > T_n^m$, then we consider the following three subcases for $n > 1$.

Sub-case 2.1: Suppose S_n^m divides u written $S_n^m | u$.

Then

$$\left\langle \frac{u - \left\langle \frac{u}{S_n^m} \right\rangle}{S_n^m - T_n^m} \right\rangle \geq \left(\frac{u - \left(\frac{u}{S_n^m} - 1 \right)}{S_n^m - T_n^m} \right) - 1$$

$$= \frac{\left(\frac{S_n^m - 1}{S_n^m} \right) u + 1}{S_n^m - T_n^m} - 1$$

$$= \left(\frac{S_n^m - 1}{S_n^m - T_n^m}\right)\frac{u}{S_n^m} + \frac{1}{S_n^m - T_n^m} - 1$$

$$\geq \frac{u}{S_n^m} + \frac{1}{S_n^m - T_n^m} - 1 \geq \frac{u}{S_n^m} - 1 = \left\langle \frac{u}{S_n^m} \right\rangle.$$

Sub-case 2.2: Suppose $S_n^m \nmid u$ and

suppose $\left(S_n^m - T_n^m\right) \nmid \left(u - \left\langle \frac{u}{S_n^m} \right\rangle\right)$

Then the notation of $\langle z \rangle$ can be replaced by $[z]$, meaning the largest integer less than or equal to z. If I is a positive integer we can use the inequalities

$$I - [z] = [I - [z]] \geq [I - z]$$

and $\frac{[z]}{I} \geq \left[\frac{z}{I}\right]$

to obtain

$$\left[\frac{u - \left[\frac{u}{S_n^m}\right]}{S_n^m - T_n^m}\right] \geq \left[\frac{\left[u - \frac{u}{S_n^m}\right]}{S_n^m - T_n^m}\right]$$

$$= \left[\frac{\left(\frac{S_n^m - 1}{S_n^m}\right)u}{S_n^m - T_n^m}\right]$$

$$= \left[\frac{(S_n^m - 1)}{(S_n^m - T_n^m)} \cdot \frac{u}{S_n^m}\right] \geq \left[\frac{u}{S_n^m}\right] = \left\langle \frac{u}{S_n^m} \right\rangle .$$

Sub-case 2.3: Suppose $S_n^m \nmid u$ but

suppose $\left(S_n^m - T_n^m\right) \mid \left(u - \left\langle \frac{u}{S_n^m} \right\rangle\right)$

Then $$\frac{u - \left\langle \frac{u}{S_n^m} \right\rangle}{S_n^m - T_n^m} = \left(\frac{u - \left[\frac{u}{S_n^m}\right]}{S_n^m - T_n^m}\right) - 1$$

Since $(S_n^m - T_n^m) \mid \left(u - \left[\frac{u}{S_n^m}\right]\right)$, we know $(S_n^m - T_n^m) \nmid \left(u - \frac{u}{S_n^m}\right)$

and furthermore $$\frac{u - \left[\frac{u}{S_n^m}\right]}{S_n^m - T_n^m} = \left[\frac{u - \frac{u}{S_n^m}}{S_n^m - T_n^m}\right] + 1$$

so $$\left\langle \frac{u - \left[\frac{u}{S_n^m}\right]}{S_n^m - T_n^m} \right\rangle - 1 = \left[\frac{u - \frac{u}{S_n^m}}{S_n^m - T_n^m} \right] + 1 - 1$$

using the same inequality employed in previous subcases,

$$\geq \left[\frac{u}{S_n^m}\right] = \left\langle \frac{u}{S_n^m} \right\rangle .$$

q.e.d.

4. HIERARCHICAL SYSTEMS CONSIDERATIONS

Some results are presented in this section for systems forming various types of hierarchies. A hierarchical system implies that a given subject must have authorized access to all objects which are authorized accessible to subjects below the given subject in the hierarchy. We will consider a ring structure, a binary symmetric tree structure, and finally, an arbitrary partial ordering.

A ring structured system is a structure such that each subject is uniquely represented by one of a number of concentric rings where any subject in an inner ring has authorized access to all objects "belonging to" any subject constituting an outer ring. However, outer ring subjects do not have authorized access to objects authorized accessible to inner ring subjects. Define $R_n^m(f)$ called the level of linear hierarchy of a mechanism, to be the maximum number of code pairs using the function f with n-bit codes and access threshold m such that all authorized accesses for a ring structure are fulfilled and no unauthorized accesses are allowed. For $f = \wedge$, $R_n^m(f) = n - m + 1$, for $f = \vee$, $R_n^m(f) = m + 1$. Appendix B documents R_n^m for various functions. Without loss of generality, we restrict our attention to systems such that one unique object belongs to each subject. The following theorem holds for these systems.

Theorem 3: Given a ring structured system (A,B,g) consisting of

(a) a set A of isolated subjects,

(b) a set B of objects, $|B| = |A| = u$, and

(c) a monotonic access mechanism within the family under consideration with $S_n^m(f) < |A|$.

Then an access code assignment satisfying all authorized accesses of the ring structure maximizes the absolute and relative degrees of protection if

(a) the assignment uses $R_n^m(f)$ minimal object codes for classes of objects,

(b) the assignment arranges $R_n^m(f)$ class codes for subjects in a minimal hierarchy, (defined in step 3 of algorithm 3 below) and,

(c) the assignment distributes subjects as evenly as possible into access classes in such a way that all subjects in a class are at consecutive levels within the hierarchy.

The major difference between this ring structure and the case of isolated subjects is the added number of authorized accesses required due to the linear ordering of subjects. Define the <u>level</u> of a subject A_i to be the number of subjects A_j (including A_i) which have authorized access to A_i, called <u>ancestors</u> of A_i. Then A_i is a <u>descendant</u> of these subjects written $A_i \leq A_j$. Note that using this terminology, the first level is 1, and all subjects are ancestors and descendants of themselves. In the case of $A_i \leq A_j$ and $A_i \neq A_j$, we write $A_i < A_j$.

The following procedure describes an algorithm which produces a maximum number of access codes and assigns subjects and their objects to access classes satisfying the constraints of a ring structure.

<u>Algorithm 3</u>:

1. Choose the first access code a_i (with i = 1) by setting the minimum number of bits in a_i such that there exists a b_i accessible to the chosen a_i.

2. Choose b_1 as the pattern with the minimum number of set bits such that
$$\sum_{k=1}^{n} f(a_{1k},b_{1k}) \geq m.$$

3. Increment i by one and set the access code a_i to a_{i-1}. Then set one more bit (chosen arbitrarily) in a_i guaranteeing the necessary condition that
$$\sum_{k=1}^{n} f(a_{ik},b_{jk}) \geq m \text{ for } j = 1, 2 \ldots, i-1.$$

   At each stage, the algorithm sets the minimum number of bits to form a ring hierarchy, so class codes for subjects are said to be arranged in a <u>minimal hierarchy</u>.

4. Choose b_i as one of the codes such that $\sum_{k=1}^{n} f(a_{ik},b_{ik}) = m$ and $\sum_{k=1}^{n} f(a_{jk},b_{ik}) < m$ for $j = 1,2 \ldots, i - 1$. This can always be done since a bit is set in a_i which was not set in any previous a_j. b_i is called a *minimal ring object code*.
5. If i is less than $R_n^m(f)$, then return to 3.
6. Order the subjects so that $A_1 < A_2 < \ldots < A_u$ where $u = |A|$. Reset i to 1.
7. Assuming the number of code pairs generated by 1 through 5 above is less than the number of subjects and divides it ($R_n^m < u$ and $R_n^m | u$), we select the u/R_n^m subjects at the highest level and assign them the access code a_i. Their objects are assigned b_1. Increment i to 2.
8. Similarly we select the u/R_n^m subjects and objects at the highest level which have not been previously selected and assign them all the codes a_i and b_i.
9. Increment i by one and repeat step 8 until all classes are filled with u/R subjects.
10. If R does not divide u, then modify steps 6 and 7 by filling all classes with [u/R] subjects except for (remainder u/R) classes which must be filled with [u/R] + 1 subjects. These can be chosen arbitrarily.

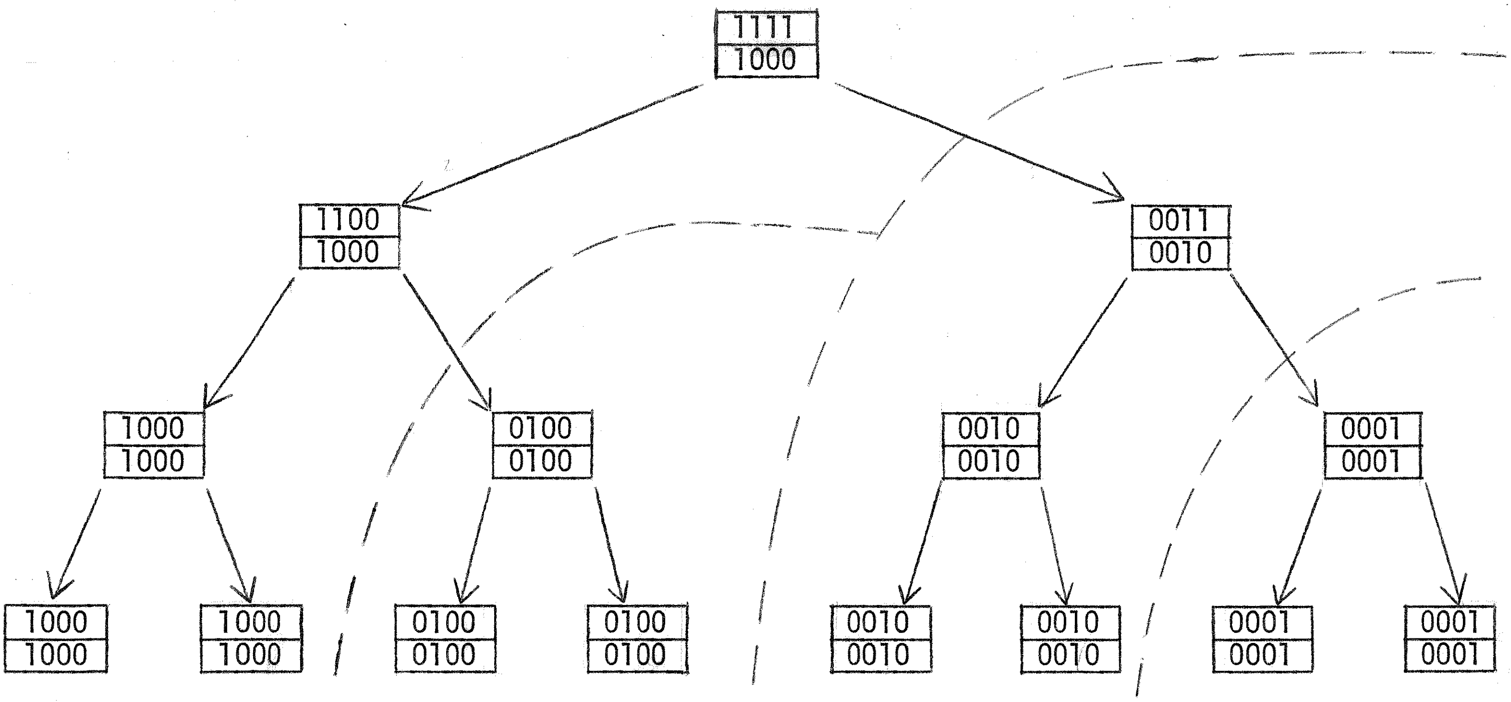
This algorithm assigns a set of consecutive subjects to each class, and thus fulfills all requirements of necessary authorized accesses. The proof that this algorithm maximizes the degree of protection is similar to the proof of theorem 1 given in the previous section. Thus the idea of the proof is only sketched informally.

Given any assignment, it is possible to permute this into an assignment satisfying the conditions of theorem 3 by first altering the number of bits set in every subject code to form a minimal hierarchy. Then alter bits of objects to form minimal ring object codes. After these alterations δ_{abs} and δ_{rel} of the system are not decreased and may be increased. If $A_i < A_j < A_k$ and A_i, A_k are assigned to one class, but A_j is assigned to another, then this is a violation of statute (c) of the theorem and the resulting assignment is non-optimal. Since A_j must have authorized access to B_i, and since $b_i = b_k$ because A_i and A_k are in the same class, we see that A_j has unauthorized access to B_k. This and all unauthorized accesses between different classes can be eliminated by making sure that all subjects in a class are at consecutive levels. Finally an argument identical to that of proof 1b can be used to establish that all classes should be of the same size, or as near to this as possible. Although not stated here as a theorem, algorithm 3 also maximizes the minimum degree of protection $\check{\delta}$ of a ring structured system.

Define a tree N to be a finite set of elements called nodes such that there is a single distinguished node called the root of the tree, and the other nodes are partitioned in $k \geq 0$ disjoint sets N_1, N_2, ..., N_k and each of these sets is in a turn a tree [12]. The trees N_1, N_2, ..., N_k are called subtrees of the root. The level of a node with respect to a tree N is one if the node is the root; otherwise it is one more than the level of the node with respect to the subtree N_i of N which contains the node. Each root is called an ancestor of the nodes in its subtrees and of itself, and conversely, each of the latter nodes is a descendant of the root. A tree structured system is one such that each subject is represented uniquely by one node of a tree. Every object in the system is associated with exactly one subject, and every subject is associated with exactly one object. In this case, the object is said to "belong to" its associated subject. Every subject of a tree structured system must have authorized access to all of its descendants' objects, but to no other nodes. Thus an object B_i is authorized accessible only to ancestors of the subject A_i to which it belongs, i.e. to those A_j such that $A \leq A_j$. In the following discourse concerning completely balanced binary tree structures we label the nodes so that the root is n_{11}. Nodes n_{21} and n_{22}, connected to n_{11} and called the direct descendants of n_{11}, are the roots of the two subtrees emanating from n_{11}. Nodes n_{31} and n_{32} are descendants of n_{11} and are the direct descendants of n_{21}; n_{33} and n_{34} are the direct descendants of n_{22}, etc. Any node n_{ij} either has no descendants or its two direct descendants are labelled $n_{i+1,2j-1}$ and $n_{i+1,2j}$. Notice that the level of a node as previously defined is its first index, $l(n_{ij}) = 1$, and all leaves (i.e. terminal nodes of the tree having no descendants) are at the same (maximal) level. The height of a tree is the maximum level of any node in the tree, $h(N) = \underset{n_{ij} \in N}{\text{Max}}\, l(n_{ij})$. Define $Q_n^m(f)$, called the <u>level of binary hierarchy</u> of a mechanism, to be the maximum number of code pairs using the function f with parameters m and n such that all authorized accesses for a symmetric binary tree are fulfilled and no unauthorized accesses are allowed. If the number of code pairs is not exactly the number needed for a totally symmetric tree, we consider a binary tree with some leaves missing as equivalent. The determination of $Q_n^m(f)$ and of an optimal access code assignment algorithm are nontrivial problems which seem to be quite dependent upon the function f as well as upon m and n. Thus we present a theorem which only gives an optimal algorithm for partitioning a tree into two access classes and then indicate that iteration of this procedure could be a basis for an assignment algorithm. After proving this theorem, a lower bound on the number of unauthorized accesses allowed by the optimal assignment is derived. This bound not only applies to trees, but to any partial ordering of subjects.

Example 4: Figure 4 shows a tree having four levels which is split in an optimal fashion into four subsets because the split minimizes the number of unauthorized accesses over all 4-way splits. Assume $f = \wedge$, $n = 4$, $m = 1$, so $Q_n^m = 4$. Each node is represented by a box containing $\begin{array}{|l|}\hline \text{subject code} \\ \hline \text{object code} \\ \hline \end{array}$.

Figure 4:



Theorem 4: Given any symmetric binary tree structured system, a partition of the objects of the tree into 2 classes maximizes the absolute and relative degree of protection of the system

iff

the partition bifurcates the tree in such a way that each subtree is wholly contained in a distinct class.

Proof 4.

The proof of theorem 4 proves that given $Q_n^m \geq 2$, and given any symmetric binary tree, the only optimal assignment of objects into two access classes is accomplished by putting the objects of the left (or right) subtree into one class and all remaining objects into the other class. In both subtrees, the access codes assigned to each class should be minimal and the corresponding subject codes should be chosen to form minimal code pairs. The object corresponding to the root of the tree is assigned to one of the two minimal codes, but the subject corresponding to the root must access all objects so it's code is not minimal. It must have a bit set to match each set bit of subjects in both subtrees which implies access to all objects as required by the tree hierarchy definition. For the access mechanism $\wedge$, with $n = 2$, $m = 1$, the above means the left subtree might have code 01, the right subtree 10 and the root subject 11.

The terminology of one node n_{ij} having access to another n'_{ij} will be used for brevity meaning that the subject corresponding to n_{ij} has access to the object corresponding to n'_{ij}. Define $x_Z(n_{ij})$ and $y_Z(n_{ij})$ to be the number of nodes in Z having authorized and unauthorized access respectively to node n_{ij}. In partitioning a symmetric binary tree as described above, refer to the larger subset by U and the smaller by W. The proof technique employed is to consider any other partitioning into subsets U', W' and show that it allows more unauthorized accesses than the U, W partition. Within any partition minimal access codes will always be assumed (except for the root subject) because setting other bits of a code can never decrease the number of unauthorized accesses and may increase it.

Case 1: $|U'| = |U|$ and $|W'| = |W|$

For any node within any subset of a partition, the number of unauthorized accessors to that node can be calculated as the total number of nodes in the subset minus the number of ancestors in the subset because ancestors have authorized access. For some partitions, this may be a lower bound because if a descendant of node n_{ij} is in a different subset of the partition and if other nodes which are not descendants of n_{ij} are also in that subset then n_{ij} has access to those other objects since they have the same access code as the descendant of n_{ij}. In partitions by our algorithm, this phenomenon never occurs. For subsets U and W, the number of unauthorized accesses can be expressed by

(4.1) $$\sum_{n_{ij}\in U} (|U| - i) + \sum_{n_{ij}\in W} (|W| - (i-1)).$$

This is true since the number of unauthorized accessors to $n_{ij}\in U$ is equal to all members of U except the ancestors of n_{ij}. There are exactly i ancestors of n_{ij} in U. If n_{ij} is in W, the same argument holds, but there are i - 1 ancestors in W. (Recall any node is an ancestor of itself). For the partition consisting of sets U' and W', a lower bound on the number of unauthorized accesses is

(4.2) $$\sum_{n_{ij}\in U'} (|U'| - x_{U'}(n_{ij})) + \sum_{n_{ij}\in W'} (|W'| - x_{W'}(n_{ij})).$$

Next we will show that the expression (4.1) is always less than the expression (4.2). That is, simplifying, we get

$$|U|^2 + |W|^2 - \sum_U i - \sum_W (i-1) < |U'|^2 + |W'|^2 - \sum_{U'} x_{U'}(n_{ij}) - \sum_{W'} x_{W'}(n_{ij}).$$

Since $|U| = |U'|$ and $|W| = |W'|$ we only need show

$$\sum_U x_U(n_{ij}) + \sum_W x_W(n_{ij}) > \sum_{U'} x_{U'}(n_{ij}) + \sum_{W'} x_{W'}(n_{ij}).$$

a. If $n_{11} \notin U'$, then $\exists 2^{h-1}$ nodes $\ni x_{U'}(n_{ij}) < i = x_U(n_{ij})$, h = height of original tree. Justification is that a node at level i has i ancestors but if n_{11} is not in the subset, the best possible is i - 1 ancestors.

b. Summing over all nodes in U' yields

$$\sum_U x_U(n_{ij}) \geq \sum_{U'} x_{U'}(n_{ij}) + |U'|.$$

c. $\exists 2^{h-1}-1$ nodes in W' $\ni x_{W'}(n_{ij}) \leq i$, but $n_{11} \notin W$ so for nodes in W, $x_W(n_{ij}) < i$.

d. Summing over all nodes in W' and W respectively yields

$$\sum_W x_W(n_{ij}) \geq \sum_W x_{W'}(n_{ij}) - |W'|.$$

e. Combining b. and d. above yields

$$\begin{aligned}\sum_U x_U(n_{ij}) + \sum_W x_W(n_{ij}) &\geq \sum_{U'} x_{U'}(n_{ij}) + \sum_{W'} x_{W'}(n_{ij}) + (|U'| - |W'|)\\ &> \sum_{U'} x_{U'}(n_{ij}) + \sum_{W'} x_{W'}(n_{ij})\end{aligned}$$

f. Suppose $n_{11} \in U'$. Since the structure of U' differs from that of U, we must have nodes of both left and right subtrees in U' and all of their ancestors also. (Otherwise U' is suboptimal.) Suppose the left subtree contains r nodes in U'. These nodes have $x_{U'}(n_{ij}) = i > x_W(n_{ij}) = i - 1$.

g. This inequity is balanced by r nodes of the right subtree $\ni n_{ij} \in U - U'$ (which must exist since $|U| = |U'|$). These nodes have $x_U = i > i - 1 \geq x_W'$. So

$$\sum_U x_U(n_{ij}) \geq \sum_{U'} x_{U'}(n_{ij})$$

h. Furthermore, all nodes of W' must have $x_{V'} \leq i - 2$ since the top nodes n_{11}, n_{21}, and n_{22} are in U'. Thus

$$\sum_W x_W(n_{ij}) > \sum_{W'} x_{W'}(n_{ij})$$

i. Combining g. and h. yields

$$\sum_U x_U(n_{ij}) + \sum_W x_W(n_{ij}) > \sum_{U'} x_{U'}(n_{ij}) + \sum_{W'} x_{W'}(n_{ij})$$

q.e.d.

Case 2: $|U'| > |U|$ and $|W'| < |W|$

In this case, use is made of the observation that $y_U(n_{ij})$ for $n_{ij} \in U$ is the same as $y_W(n_{ik})$ for $n_{ik} \in W$. This is true because

$|U| - x_U(n_{ij}) = |U| - i = (|U| - 1) - (i - 1) = |W| - x_W(n_{ik})$. The only requirement is that n_{ij} and n_{ik} be at the same level. Thus we will refer to $y(n_{ij})$ as the number of unauthorized accessors irrespective of whether $n_{ij} \in U$ or $n_{ij} \in V$.

a. $|U'| > |U| \Rightarrow y_{U'}(n_{ij}) = |U'| - x_{U'}(n_{ij}) \geq |U'| - i > |U| - i = y(n_{ij}) \forall n_{ij} \in U'$.

b. Since $|U'| = |U| + k$ for some integer $k > 0$, $y_{U'}(n_{ij}) \geq y(n_{ij}) + k$. Thus

$$\sum_{n_{ij} \in U'} y_{U'}(n_{ij}) \geq \sum_{n_{ij} \in U'} y(n_{ij}) + k|U'|.$$

c. $|U'| = |U| + k$ implies $|W'| = |W| - k$. So $n_{ij} \in W'$ may have k less unauthorized accessors than if it were in W or U, but no less than this, $y_{W'}(n_{ij}) \geq y(n_{ij})$ for all n_{ij} in W'. Thus

$$\sum_{n_{ij} \in W'} y_{W'}(n_{ij}) \geq \sum_{n_{ij} \in W'} y(n_{ij}) - k|V'|$$

d. Combining b. and c. yields

$$\sum_{U'} y_{U'}(n_{ij}) + \sum_{W'} y_{W'}(n_{ij}) \geq \sum_{U' \cup W'} y(n_{ij}) + k|U'| - k|W'|$$

e. Since $k(|U'| - |W'|) > 0$ and $U' \cup W' = U \cup W$, we get

$$\sum_{U'} y_{U'}(n_{ij}) + \sum_{W'} y_{W'}(n_{ij}) > \sum_{U} y_U(n_{ij}) + \sum_{W} y_W(n_{ij}).$$

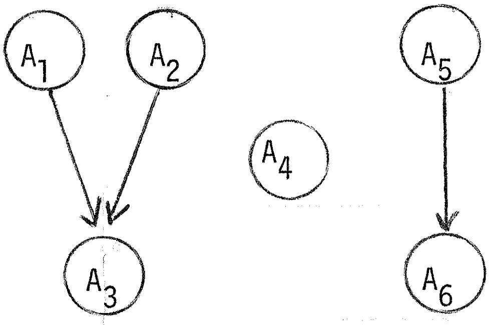
q.e.d.

Notice that in splitting the subtrees in figure 4, the left subtree again fits the criterion of theorem 4, but the right subtree has a chain of two nodes in place of the root. The proof 4. also holds in the general case of the root consisting of a ring structure (i.e. a chain of nodes) if the chain is not too long. Thus the algorithm for partitioning a tree into an arbitrary number of subsets might consist of repeated applications of theorem 4 plus a method of assigning codes to access classes. One question which arises in this more general case is whether to put all nodes of the chain into a single subset when splitting. It appears that all nodes should be lumped together as the root and put into one class to avoid having a descendant of one of the root nodes n_{ij} in a different set from n_{ij}, but in the same class as other nodes which are not descendants of n_{ij}. This causes unauthorized accesses across subset boundaries and eventually implies a non-optimal algorithm.

The algorithm presented is, in general, non-optimal and could be improved by (among other things) checking at each step that the longest chain has not gotten so long that separating it from its lowest node and tree is not more efficient than breaking a tree into its subtrees.

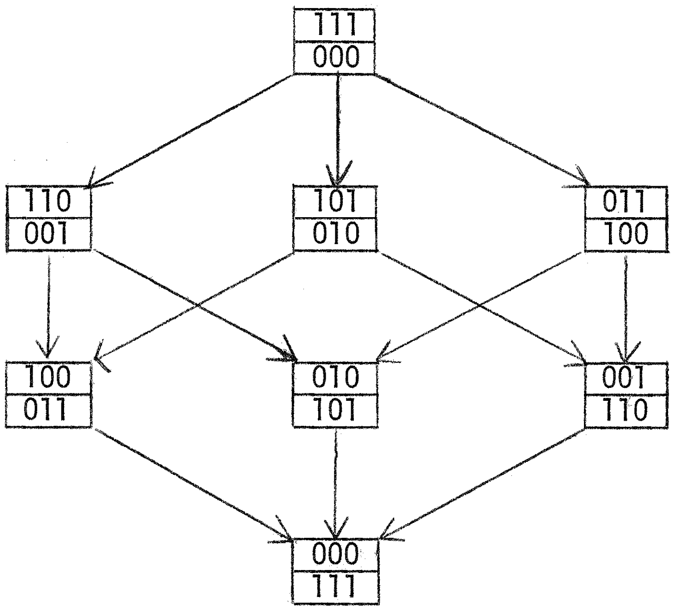
In these last paragraphs, a lower bound will be derived for the number of unauthorized accesses for an arbitrary hierarchy. For this derivation, we can drop the restrictions of a monotonic access mechanism and of being restricted to setting bits within a given field length. In its most general form, we mean a hierarchy to be any partial order. Thus, the structure given in figure 5a is a hierarchy.

figure 5a:



Different access mechanisms can do quite well on different specialized types of hierarchies. Thus f = V, n=m=3 fits the hierarchy of figure 5b very well although $S_3^3(V) = 3$.

figure 5b:



Each box $\begin{array}{|c|}\hline a_i \\ \hline b_i \\ \hline\end{array}$ represents the codes of a subject and its object associated with the node.

Theorem 5: Given n nodes arranged as an arbitrary hierarchy and d possible code words. Then if $n \geq d$, there must be at least $\frac{n}{2}(\gamma-1)$ unauthorized accesses in the system, where $\gamma = \left[\frac{n}{d}\right]$.

Proof 5:

For each code word, we can form an access class. At best, an access class containing γ_i nodes may have no unauthorized accessors from outside of the class, and a minimum number of unauthorized accessors from within the class. This minimum is obtained if the nodes in the class form a ring structure so one (the highest level) object has no unauthorized accessors, because all accesses are authorized, another (the second highest) has one unauthorized accessor, etc., until the lowest level node which has $\gamma_i - 1$ unauthorized accessors. The total number of unauthorized accesses within the class is the sum of these

(4.3) $$\sum_{j=0}^{\gamma_i - 1} j = \frac{\gamma_i(\gamma_i - 1)}{2}$$

Summing over all classes gives the constraint

(4.4) $$\sum_{i=1}^{d} \gamma_i = n$$

Minimizing the sum of expressions of type 4.3 is equivalent to omitting the denominators and minimizing $\sum_{i=1}^{d} \gamma_i(\gamma_i - 1)$ which was previously shown to have a solution of $\gamma_i = n/d$, i = 1, 2 ..., d. Define $\gamma = \langle n/d \rangle$ where $\langle z \rangle$ means the largest integer not greater than z. Then the total number of accesses in any hierarchical system is bounded from below by

$$d\left(\frac{n}{d}\left(\frac{\gamma - 1}{2}\right)\right) = \frac{n}{2}(\gamma - 1).$$

q.e.d.

5. FINAL CONSIDERATIONS

In conclusion, we have defined access mechanisms as functions which are evaluated to determine if subjects have access to objects. A family of mechanisms which includes a number of well-known systems was pinpointed for investigation. The utility of various mechanisms must, of course, depend upon the assignment of access codes to subjects and objects. The particular semantics of the system (i.e. which subjects are authorized to access which objects) must determine how these codes are assigned. Cases considered in this paper include isolated subjects, ring and tree structures, and arbitrary hierarchies. The situation of more subjects than access classes was considered and some algorithms were presented which assign access codes to maximize the degree of protection in this situation.

A number of refinements and extensions of the concepts presented are possible directions of future research. This study has been concerned with static assignment algorithms. An extension to this is an environment in which the population of subjects and objects is dynamically changing. A subject could upon entering the system specify the degree of security that he demands and is willing to pay for. Another refinement is to distinguish between subjects, each having a reliability factor attached to it (undebugged modules = low reliability, "proved" modules = high reliability), and re-define the degree of security to take this factor into account. The general access mechanism defined in this paper suggests that there are design alternatives that should be studied before implementation. Consideration of dynamic assignment algorithms leads to the possibility of new hardware mechanisms allowing dynamic alteration of the access threshold m. An intriguing open problem is to discover an optimal general assignment algorithm for tree structured systems. Finally, many other functions than the family of access mechanisms investigated in this study could be examined (e.g. $\sum_{k=1}^{n} f(a_{ik},b_{jk}) = m$). Indeed, it is not necessary that all bits be weighted equally; A. Ehrenfeucht has suggested [3] that the family of weighted functions, $\sum_{k=1}^{n} w_k f(a_{ik},b_{jk})$ might be useful and amenable to mathematical analysis. In the field of computer security and protection, there is a preponderance of qualitative results and a dearth of quantitative results. Thus it is hoped that the spirit and content of this paper will contribute to and encourage further development of a "Theory of Protection".

Acknowledgments

The author would like to extend his grateful appreciation to Professor G. J. Nutt for feed-back (and feed-forward) during many stimulating discussions of access mechanisms and protection in the MAP system from which ideas presented in this paper evolved.

Clarence A. Ellis

APPENDIX A

In this appendix, it is shown that the values of $S_n^m(f)$ and $R_n^m(f)$ used in the paper really are the maximum number of object codes allowing no unauthorized accesses in a system of isolated subjects and of ring structured subjects respectively. Furthermore, it is shown that no more than n(f) bits need be considered (n(f) = field length). The exposition is restricted to consideration of functions AND(Λ) and OR(V). Justifications for the other monotonic boolean functions can be derived from these two functions. Appendix B gives a table of values of S_n^m and R_n^m for all monotonic boolean functions. Thus it is not difficult to assign codes to subjects if $|A|$ is less than these maxima. The more difficult task of assigning codes if $|A|$ is greater than the maximum is the subject of the main body of this paper.

For the function Λ, S_n^m describes the number of access code pairs (a_i, b_i) such that $\sum_{k=1}^{n} (a_{ik} \Lambda b_{ik}) \geq m$ but $\sum_{k=1}^{n} (a_{ik} \Lambda b_{jk}) < m$ for all i and all $j \neq i$. By choosing exactly m bits to set to one in constructing a_i and then constructing $b_i = a_i$, $\Sigma(a_{ik} \Lambda b_{ik}) \geq m$ is obtained. Another (a_j, b_j) can be constructed by choosing a different but possibly overlapping set of m bits. The number of ways of choosing m bits out of n is the number of combinations $\binom{n}{m}$. Notice that no a_i can access any b_j if $i \neq j$. Finally we argue that increasing any code c_i to more than m set bits can never help our cause because of the monotonicity of our function. In using all combinations of m bits as codes, every bit position is set in some code, so the minimum field length is n.

For the function V, it is sufficient to have $k \leq m$ bits set in a_i, and m-k (non-overlapping) bits set in b_i so that $\Sigma(a_{ik} \text{ V } b_{ik}) \geq m$. Due to the monotonicity of the function, it is again useless to have more than a total of m bits set, i.e. $\alpha_i + \beta_i \not> m$. The number of ways of choosing k bits out of m, $\binom{m}{k}$, is the total number of codes for the given k. This total is maximized when k is as close an integer value as possible to m/2, so $S_n^m(V) = \binom{m}{\left[\frac{m}{2}\right]}$. In this case it is possible to assign all codes so that only m bits are used and the remaining n-m bits are not set in any of the codes a_i or b_i. This assertion is stated and proved as a theorem.

Theorem A: Given a system (A,B,g) consisting of

(a) a set A of isolated subjects

(b) a set B of objects such that $|B| = |A|$

(c) a monotonic access function of the form

$$\sum_{k=1}^{n}(a_{ik} \vee b_{jk}) \geq m \text{ where } S_n^m(V) = |A|$$

Then there exists an access code assignment F which yields a degree of security (absolute and relative) of one, and which has $|F| = m$ as the minimum field length. The theorem implies that there is an assignment which uses (i.e. sets in all subjects and objects) no more than m bits. Degree of security one implies a totally secure system: $\delta_{abs} = \delta_{rel} = 1 \Rightarrow \bar{y} = 0 \Rightarrow$ no unauthorized accesses. As stated above, it is possible to obtain $\binom{m}{\left[\frac{m}{2}\right]} = S_n^m(V)$ isolated code pairs. From this, the proof shows that bits outside of the field can be moved inside, one-by-one, without decreasing the degree of security.

<u>Proof A</u>:

The discussion before Theorem A established that there exists an assignment such that all codes form isolated minimal code pairs and the number of isolated pairs is $S_n^m(V)$. If this assignment has a field length of m then we're finished. If not, then pick m of the largest sums $e_k = \sum_{i=1}^{|A|} a_{ik}$ from k = 1,2,...n as the bits constituting the field. If possible pick an object which has a bit set outside of the field, reset all bits outside of the field and set bits inside of the field which can be done by the existence and uniqueness of code pairs. If bit ℓ outside the field was reset, then this caused e_ℓ-d new unauthorized accesses, and setting bit j inside the field deducts e_j-d unauthorized accesses where $d = \sum_{i=1}^{|A|}(a_{ij} \wedge a_{i\ell})$. By choice of field we know $e_j \geq e_\ell$ so the net change in unauthorized accesses is $(e_\ell - d)-(e_j - d) \leq 0$. This implies that the move hasn't degraded the system protection, and repetition of the above step puts all b_i within the field. Now moving the subjects inside of the field is easy because a move will not cause any new unauthorized accesses. Repetitions of these steps yields all access codes with bits set only within the field.

q.e.d.

The value of $R_n^m(f)$ is defined as the maximum number of access code pairs (a_i, b_i) which can be constructed in such a manner that each a_i can access b_j if and only if $j \leq i$. These codes can be constructed by assigning to a_1 any code which has a minimum number of bits set to still allow construction of a b_i which is accessible. Thus $f = \wedge$ implies a_1 has m bits set to one; f = V implies a_1 has zero bits set to one. Further subject codes are derived by setting one bit at a time as described in algorithm 3. For $f = \wedge$, the maximum number of bits possibly set is n, implying a total number of codes of $R_n^m(\wedge) = n-m+1$. For f = V, the maximum number is m because creating other subjects with more than m bits set would cause unauthorized accesses. This implies the total number of subject codes is $R_n^m(V) = m + 1$.

APPENDIX B

Table of values of S_n^m and R_n^m for all monotonic functions

function →	n ↓	m ↓	0		1		a		b		$\wedge$		$\vee$		$\bar{\wedge}$		$\bar{\vee}$	
	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1	0	0	1	1	1	1	1	1	1	1	1	2	1	2	1	1
	2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1	0	0	1	1	1	1	1	1	2	2	1	2	1	2	2	2
		2	0	0	1	1	1	1	1	1	1	1	2	3	2	3	1	1
	3	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1	0	0	1	1	1	1	1	1	3	3	1	2	1	2	3	3
		2	0	0	1	1	1	1	1	1	3	2	2	3	2	3	3	2
		3	0	0	1	1	1	1	1	1	1	1	3	4	3	4	1	1
	4	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		1	0	0	1	1	1	1	1	1	4	4	1	2	1	2	4	4
		2	0	0	1	1	1	1	1	1	6	3	2	3	2	3	6	3
		3	0	0	1	1	1	1	1	1	4	2	3	4	3	4	4	2
		4	0	0	1	1	1	1	1	1	1	1	6	5	6	5	1	1
arbitrary →	n	m	0	0	1	1	1	1	1	1	$\binom{n}{m}$	n-m+1	$\binom{m}{\left[\frac{m}{2}\right]}$	m+1	$\binom{m}{\left[\frac{m}{2}\right]}$	m+1	$\binom{n}{m}$	n-m+1
			S_n^m	R_n^m	S_n^m	R_n^m	S_n^m	R_n^m	S_n^m	R_n^m	S_n^m	R_n^m	S_n^m	R_n^m	S_n^m	R_n^m	S_n^m	R_n^m

Notice that the functions pairs $(\wedge, \overline{\vee})$ and $(\vee, \overline{\wedge})$ have identical values. This can be expected by duality of boolean functions. Similar expectations hold for the pairs $(a, \overline{a})$ and $(b, \overline{b})$, so the functions $\overline{a}$ and $\overline{b}$ were omitted from the table. The function $\equiv$ is not a monotonic function. As far as the author could ascertain, no closed formula is known for $S_n^m(\equiv)$ although it forms a distance function and an upper bound on it is given by the Hamming bound [1].

REFERENCES

1. Ash, R. B., *Information* Theory, Interscience Publishers (New York, N.Y.) 1965.

2. Carroll, J. M.; and McLellan, P. M. "The Data Security Environment of Canadian Resource-Sharing Systems" INFOR 9, 1(March 1971) pp. 58-67.

3. Ehrenfeucht, A; Private communications.

4. Graham, G. S. and Denning, P. J., "Protection-Principles and Practice", Proceedings AFIPS 1972 SJCC, vol. 40, AFIPS Press (Montvale, N. J.) pp. 417-429.

5. Graham, R. M. "Protection in an Information Processing Utility" *Communications of the ACM*, 11, 5(May 1968) pp. 365-369.

6. Hansen, P. B. (Ed.) *RC-4000 Software Multiprogramming System*, A/S Regnecentralen, Copenhagen, February 1971.

7. Hoffman, L. J., "Computers and Privacy: A Survey" *Computing Surveys* 1, 2 (June 1969), pp. 85-104.

8. Hoffman, L. J., *The Formulary Model for Access Control and Privacy in Computer Systems*, Ph. D. Thesis, Stanford Linear Accelerator Center, Stanford University (SLAC-117) May 1970.

9. IBM Operating System/360, Concepts and Facilities, Document C28-6535.

10. Jones, A. K., *Protection in Programmed Systems*, Ph.D. Thesis, Carnegie-Mellon University, Department of Computer Science, June 1973.

11. Katzan, H., *Computer Data Security*, Van Nostrand Reinhold (Cincinnati, Ohio) 1973.

12. Knuth, D. E., *The Art of Computer Programming*, vol. 1, Addison-Wesley, Reading, Mass.) 1969.

13. Lampson, B. W., "Protection" Proceedings Fifth Annual Princeton Conference on Information Sciences and Systems, Princeton University, March 1971, pp. 437-443.

14. Nutt, G. J., "Multi Associative Processor Evaluation Study," MAP memo no. 1, University of Colorado, Department of Computer Science, November 1973.

15. Palme, J. "Software Security" *Datamation*, January 1974, pp. 51-55.

16. Peters, B., "Security Considerations in a Multiprogrammed Computer System," Proceedings AFIPS 1967 SJCC, vol. 30, AFIPS Press (Montvale, N. J.) pp. 283-286.

17. Shroeder, M. D., and Saltzer, J. H. "A Hardware Architecture for Implementing Protection rings", *Communications of the ACM*, 15, 3(March 1972) pp. 157-170.

18. Weiss, H. "Computer Security, An Overview", *Datamation*, January 1974, pp. 42-47.

19. Weissman, C. "Trade-off Considerations in Security System Design" *Seminar on Privacy*: Legal and Technical Protection in the Computer Age, October 1970, 13 pages.