AD-A062 003 DEPARTMENT OF DEFENSE WASHINGTON D C F/G 15/3
OPTIMIZATION OF A COMPUTER SECURITY INDEX VERSUS COST,(U)
JUN 78 R P WISSING

UNCLASSIFIED NL

1 OF 1
AD A062003

END
DATE FILMED
3-79
DDC

LEVEL II

(2)

(6) Optimization of a Computer Security Index Versus Cost

(10) Richard P. Wissing
Dept. of Defense

(11) June 1978

(12) 67 p.

109050

-A-

ABSTRACT

In this paper, we propose a computer security index for measuring the security of computer systems and a strategy for purchasing computer security countermeasures in a cost effective manner. Required inputs for the model include definition of threats and countermeasures, relative importance of threats, costs of countermeasures, and the effectiveness of each countermeasure against each of the threats listed. If a standardized list of threats and countermeasures can be developed, the computer security index could also be used to compare the security of different computer systems.

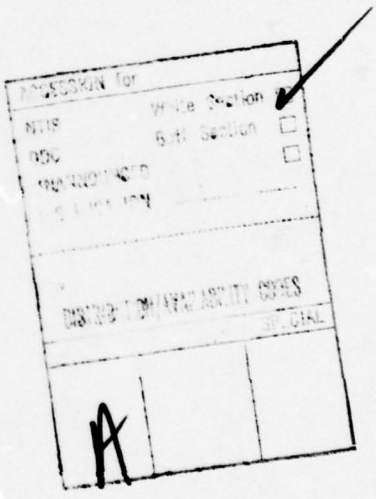


TABLE OF CONTENTS

2. Questions to be Addressed

In this paper, we provide answers to questions such as those listed below provided that the inputs listed below are given.

Question 1: We are making a one-time purchase of countermeasures for a particular computer system. For a fixed budget allocation, which set of countermeasures yields the most security per dollar?

Question 2: We will be purchasing countermeasures over an extended period of time. Which countermeasures should we buy now and which countermeasures should be purchased later?

Inputs required for Questions 1 and 2:

a. Definition of threats and countermeasures

b. Relative importance of threats

c. Costs of countermeasures

d. Effectiveness of each countermeasure against each threat listed.

Question 3: We are considering two computer systems. Which system is the most secure?

Inputs required for Question 3:

Inputs a, b, c, and d above with the added provision that there is a standard list of threats and countermeasures for the two systems.

There are other questions which could be answered by the model and the accompanying computer program. These questions would generally be different versions, or rearrangements, of the above questions. Two such sample questions might be:

Question 4: We have several countermeasure packages which we can purchase on an all or none basis. Which package is the most secure?

Question 5: We have already selected several sequences for purchasing countermeasures and plan to implement one of them. Which sequence is best from a security standpoint?

As with Questions 1 and 2, the inputs required to answer Questions 4 and 5 are inputs a, b, c, and d above.

Given the flexibility of the model and the accompanying computer program, one might be able to answer other questions relating to the security of computer systems, or for that matter, be able to answer questions relating to the security of other systems which are not computer systems.

3. Discussion of the Methodology

In this section, we shall discuss (1) the model we have developed - applicable to all five questions above; (2) the measure of computer security we have developed - called the computer security index - applicable to all five questions above, but particularly applicable to Questions 3, 4 and 5; and (3) the algorithms we have developed (applicable to Questions 1 and 2 above).

3.1. The Model

We have the following situation. An adversary has certain objectives in attacking a computer, e.g., denying use by an authorized user, obtaining access to classified or privileged information, etc. We label these, say K objectives, O_1, O_2, . ., O_K, and assign them relative weights w_1, w_2, . . ., w_K. To achieve these objectives, the adversary can utilize any of certain threats, say threats T_{k1}, T_{k2}, . . ., T_{kI_k} for objective O_k, k = 1, 2, . . ., K. It is quite possible that the same threat could achieve two or more different objectives. For our purposes, we would list the threat twice, calling it, say T_{12} and T_{24}, if it were the second threat listed to achieve objective O_1 and the fourth threat listed to achieve objective O_2. We also have some countermeasure set CM_1, CM_2, . . ., CM_n which helps prevent the adversary from carrying out the threats to achieve his objectives. We know the cost of the countermeasures and we have a measure of effectiveness of each countermeasure CM_j against each threat T_{ki} in achieving objective O_k - calling this measure of effectiveness C_{kij}. This measure of effectiveness C_{kij} represents, mathematically, the probability of countermeasure CM_j stopping threat T_{ki} against objective O_k. Diagrammatically we have the situation depicted in Figure 1 (page 7).

The case where there is a one-to-one correspondence between threats and objectives, that is, where $I_1 = I_2 = . . . = I_K = 1$, will be called the simplified model. And the case where there is a many-to-one correspondence between threats and objectives, that is, where some $I_k \neq 1$, k = 1, 2, . . ., K, will be called the generalized model.

FIGURE 1

			COUNTERMEASURE		
			CM_1	CM_2 . . .	CM_n
Objective O_1	Threat	T_{11}	C_{111}	C_{112} . . .	C_{11n}
		T_{12}	C_{121}	C_{122} . . .	C_{12n}
		.	.		.
		.	.		.
		.	.		.
		T_{1I_1}	C_{1I_11}	C_{1I_12}. . .	C_{1I_1n}
Objective O_2	Threat	T_{21}	C_{211}	C_{212} . . .	C_{21n}
		T_{22}	C_{221}	C_{222} . . .	C_{22n}
		.	.	.	.
		.	.	.	.
		.	.	.	.
		T_{2I_2}	C_{2I_21}	C_{2I_22}. . .	C_{2I_2n}
		.			
		.			
		.			
Objective O_K	Threat	T_{K1}	C_{K11}	C_{K12}	C_{K1n}
		T_{K2}	C_{K21}	C_{K22} . . .	C_{K2n}
		.	.	.	.
		.	.	.	.
		.	.	.	.
		T_{KI_K}	C_{KI_K1}	C_{KI_K2}	C_{KI_Kn}

3.2. The Computer Security Index

We propose the following computer security index U' for the full countermeasure set (CM_1, CM_2, . . ., CM_n).

$$U' = \sum_{k=1}^{K} w_k \prod_{i=1}^{I_k} \left[1 - \prod_{j=1}^{n} (1 - C_{kij})\right]$$

Let us examine this function:

C_{kij}: represents probability that countermeasure CM_j stops threat T_{ki} against objective O_k.

$\bar{C}_{kij} = (1-C_{kij})$: represents probability that countermeasure CM_j does not stop threat T_{ki} against objective O_k.

(Continued on next page.)

$\bar{C}_{ki\cdot} = \prod_{j=1}^{n} (\bar{C}_{kij})$: represents probability that threat T_{ki} is not stopped against objective O_k.

$C_{ki\cdot} = 1 - \bar{C}_{ki\cdot}$: represents probability that threat T_{ki} against objective O_k is stopped.

$C_{k\cdot\cdot} = \prod_{i=1}^{I_k} C_{ki\cdot}$ - represents probability that all threats against objective O_k are stopped, i.e., probability that objective O_k is protected. Recalling that w_k represents the relative weight of objective k, U' then equals $\sum_{k=1}^{K} w_k C_{k\cdot\cdot}$.

At this point, several observations may be made. First, we have that $0 \leq U' \leq 1$, if we replace w_k by $W_k = \frac{w_k}{\sum_{t=1}^{K} w_t}$. So call this new index U where we replace w_k by W_k in the definition of U'. Second, we have $C_{k\cdot\cdot} \leq C_{ki\cdot}$ for all $i = 1, 2, \ldots, I_k$. This is a "weakest link" type property - that is, the probability of protecting an objective is no higher than the lowest probability of stopping any particular threat against that objective. In particular, this means that if we have zero probability of stopping threat T_{ki} against objective O_k, we have zero probability of protecting objective O_k. This all seems reasonable since any adversary would certainly choose a threat with the least resistance to achieve an objective.

If we had something less than a full countermeasure set, the computer security index would be:

$$U = \sum_{k=1}^{K} W_k \prod_{i=1}^{I_k} \left[1 - \prod_{j=1}^{n} (1 - C_{kij})^{x_j}\right]$$

where $x_j = 1$ if countermeasure CM_j is included in set

0 if countermeasure CM_j is not included in set

3.3. The Algorithms

We can represent the countermeasure set being used by an n-long binary vector

$$Y = (x_1, x_2, \ldots, x_n)$$

where: $x_j = 1$ if countermeasure CM_j is included in set Y

0 if countermeasure CM_j is not included in set Y

So, for n = 5, the vector Y = (1 0 1 1 0) means countermeasures CM_1, CM_3, and CM_4 are included in the countermeasure set and CM_2 and CM_5 are not included. There are of course 2^n possible countermeasure sets. Our goal is to develop a particular sequence of countermeasure sets $Y_1, Y_2, \ldots, Y_n$ that is in some sense cost effective, where Y_i and Y_{i+1} differ only in that one position in the vector has a zero in Y_i and a one in Y_{i+1}. For example, if

$$Y_i = (0\ 1\ 1\ 1\ 0\ 1)$$

the only candidates for Y_{i+1} are

$$Y_{i+1} = (1\ 1\ 1\ 1\ 0\ 1)$$

and

$$Y_{i+1} = (0\ 1\ 1\ 1\ 1\ 1).$$

So we start with an empty countermeasure set and add one countermeasure at a time, in some cost effective manner until we have a full countermeasure set.

For each countermeasure set Y_i, we have an associated computer security index U_i and an associated cost $E_i = \sum_{j=1}^{n} D_j x_j$ where D_j is the cost of countermeasure CM_j. We could plot U_i vs. E_i and for a budget E_i, we choose countermeasure set Y_i. If we get an additional amount of money $E_{i+1} - E_i$, we use countermeasure set Y_{i+1}, obtained by adding one countermeasure to countermeasure set Y_i.

Let us now consider several algorithms for developing the sequence of countermeasure sets to be purchased. For notational purposes, let U_i^j, $i = 1, 2, \ldots, n$, $j \in (1,2,\ldots,n)$ be the computer security index for countermeasure set Y_{i+1} obtained by adding countermeasure CM_j to countermesure set Y_i. We start with $Y_0 = (0, 0, \ldots, 0)$.

In our first algorithm G1, we choose countermeasure CM_j which maximizes $\dfrac{U_i^j}{E_i + D_j}$. That is, we choose countermeasure CM_j which maintains the highest security to cost ratio. After choosing countermeasure CM_j, we have $E_{i+1} = E_i + D_j$, and $U_{i+1} = U_i^j$.

In another algorithm, call it G2, we choose countermeasure CM_j which maximizes $\dfrac{U_i^j - U_i}{D_j}$. That is, we choose countermeasure CM_j which gives the most increase in security per dollar. After choosing countermeasure CM_j, we have $U_{i+1} = U_i^j$ and $E_{i+1} = E_i + D_j$.

The above algorithms are workable in the case $I_1 = I_2 = \ldots = I_k = 1$ which is the case where there is a 1-1 correspondence between threats and objectives. In the more general case, however, where there is a many-to-one correspondence between threats and objectives, (i.e., some $I_k \neq 1$, $k = 1, 2, \ldots, K$), we may run into immediate trouble if we try to use algorithms G1 or G2 since it is quite possible that $U_i^j = U_i$ for all $j = 1, 2, \ldots, n$. In such a case, we could not determine which countermeasure would be best to add.* To counter this effect, we propose a sort of "reverse G2" which we will label $\overline{G2}$. Under this algorithm we start with a full countermeasure set $Y_n = (1, 1, \ldots, 1)$ and delete countermeasures so that the decrease in security index per dollar is minimized.

*Algorithm G1 would add the least costly countermeasure regardless of its impact on security. Algorithm G2 would arbitrarily add the countermeasures in numerical sequence. Neither method is very desirable.

That is, if we let $\bar{U}_i^j$ represent the index for countermeasure set Y_{i-1} obtained by deleting countermeasure CM_j from countermeasure set Y_i, we delete countermeasure CM_j such that $\frac{U_i - \bar{U}_i^j}{D_j}$ is minimized. After deleting counteremasure CM_j, we have $U_{i-1} = \bar{U}_i^j$ and $E_{i-1} = E_i - D_j$. This gives us a sequence of countermeasure sets Y_n, Y_{n-1}, ..., Y_1 which is the reverse of the sequence we desire.

(continued on next page)

We could also perform a "reverse G1", call it $\overline{G1}$, where we delete countermeasure CM_j such that $\overline{U_i^j}$ is maximized at each step. That is,

$$E_i \overline{-D_j}$$

at each step, we delete a countermeasure so as to maintain the highest security to cost ratio possible.

In Section 5 we will examine the performance of the four algorithms for sample sets of data. In general, we recommend applying all applicable algorithms and choosing the one which gives the best results for the questions being addressed.

We should note in passing that none of the algorithms proposed above guarantee an optimal solution. Determining the optimal solution would require that all n! possible sequences or all 2^n possible countermeasure sets be examined - a task which could easily become prohibitive.

4. Toy Problem (with Sample Calculations)

To illustrate the calculations required for the computer security index and the algorithms, let us consider the following toy problem. Suppose we have three objectives (encompassing five threats) with the relative and normalized weights listed below.

Objective	Threats	Relative Weight	Normalized Weight
O_1	T_{11}	3	.200
	T_{12}		
O_2	T_{21}	5	.333
O_3	T_{31}	7	.467
	T_{32}		

(Since there is a 5 to 3 correspondence between threats and objectives this is an example of the generalized model.)

Assume we have five countermeasures with the following costs:

Countermeasure	Cost
CM_1	10
CM_2	10
CM_3	5
CM_4	3
CM_5	1

Our countermeasure effectiveness matrix, representing the probability C_{kij} of countermeasure CM_j blocking threat T_{ki} against objective O_k is assumed to be:

		Countermeasures				
		CM_1	CM_2	CM_3	CM_4	CM_5
THREATS	T_{11}				.6	.9
	T_{12}		.9		.6	
	T_{21}			.5		
	T_{31}	.8				
	T_{32}		.7			

(The computer program input file for this toy program is found in Appendix II, Table II-1.)

Let us now execute algorithm $\overline{G}2$ making use of the notation of Section 3. We must first calculate the computer security index for the full countermeasure set. The numbers of threats in each objective are $I_1 = 2$, $I_2 = 1$, and $I_3 = 2$, and the only non-zero C_{kij} are:

$C_{114} = .6$ $C_{122} = .9$ $C_{213} = .5$

$C_{115} = .9$ $C_{124} = .6$ $C_{311} = .8$

$C_{322} = .7$

Other quantities and their values follow:

$\overline{C}_{ki}$. = Probability that threat T_{ki} is not stopped against objective O_k.

$\overline{C}_{11}$.= (1 - .6) (1 - .9) = .04

$\overline{C}_{12}$.= (1 - .9) (1 - .6) = .04

$\overline{C}_{21}$.= (1 - .5) = .5

$\overline{C}_{31}$.= (1 - .8) = .2

$\overline{C}_{32}$.= (1 - .7) = .3

C_k.. = Probability that all threats against objective O_k are stopped

$C_1.. = (.96)\ (.96) = .922$

$C_2.. = .5$

$C_3.. = (.8)\ (.7) = .56$

The computer security index of the 5 countermeasures then equals

$U_5 = (.2)\ (.922) + (.333)\ (.5) + (.467)\ (.56) = .612$

We represent the full countermeasure set by $Y_5 = (1\ 1\ 1\ 1\ 1)$ and its cost by $E_5 = 29$.

To continue with iteration 1 of the algorithm, we must calculate $\overline{U}_i^j$ and F_i^j, $j = 1,2,3,4,5$ for $i = 5$ where $\overline{U}_i^j$ is the index if countermeasure CM_j is deleted from countermeasure set Y_i and F_i^j is the factor we are trying to minimize. For algorithm $\overline{G}2$, $F_i^j = \dfrac{U_i - \overline{U}_i^j}{D_j}$ where D_j is the cost of countermeasure CM_j and U_i is the index of countermeasure set Y_i. Some of the detailed calculations for the first iteration are shown below.

Delete CM_1

$\bar{c}_{11}. = .04 \quad C_1.. = (.96)\ (.96) = .922$

$\bar{c}_{12}. = .04$

$\bar{c}_{21}. = .5 \quad C_2.. = .5$

$\bar{c}_{31}. = 1.0 \quad C_3.. = (0)\ (.7) = 0$

$\bar{c}_{32}. = .3$

$\overline{U}_5^1 = (.2)\ (.922) + (.333)\ (.5) + (.467)\ (0) = .351$

$F_5^1 = \dfrac{.612 - .351}{10} = .026$

Delete CM_2

$\bar{C}_{11.} = .04$ $\quad C_{1..} = (.96)(.6) = .576$

$\bar{C}_{12.} = (1.0)(.4) = .4$

$\bar{C}_{21.} = .5$ $\quad C_{2..} = .5$

$\bar{C}_{31.} = .2$ $\quad C_{3..} = (.8)(0) = 0$

$\bar{C}_{32.} = 1.0$

$\bar{U}_5^2 = (.2)(.576) + (.333)(.5) + (.467)(0) = .282$

$F_5^2 = \frac{.612 - .282}{10} = .033$

(Continued on next page.)

Delete CM_3

$\bar{C}_{11.} = .04$ $\quad C_{1..} = .922$

$\bar{C}_{12.} = .04$

$\bar{C}_{21.} = 1.0$ $\quad C_{2..} = 0$

$\bar{C}_{31.} = .2$ $\quad C_{3..} = .56$

$\bar{C}_{32.} = .3$

$\bar{U}_5^3 = (.2)\ (.922) + (.333)\ (0) + (.467)\ (.56) = .446$

$F_5^3 = \frac{.612 - .446}{5} = .033$

Delete CM_4

$\bar{C}_{11.} = (1.0)\ (.1) = .1$ $\quad C_{1..} = (.9)\ (.9) = .81$

$\bar{C}_{12.} = (.1)\ (1.0) = .1$

$\bar{C}_{21.} = .5$ $\quad C_{2..} = .5$

$\bar{C}_{31.} = .2$ $\quad C_{3..} = .56$

$\bar{C}_{32.} = .3$

$\bar{U}_5^4 = (.2)\ (.81) + (.333)\ (.5) + (.467)\ (.56) = .590$

$F_5^4 = \frac{.612 - .590}{3} = .007$

Delete CM_5

$\bar{C}_{11.} = (.4)\ (1.0) = .4$ $\quad C_{1..} = (.6)\ (.96) = .576$

$\bar{C}_{12.} = .04$

$\bar{C}_{21.} = .5$ $\quad C_{2..} = .5$

$\bar{C}_{31.} = .2$ $\quad C_{3..} = .56$

$\bar{C}_{32.} = .3$

$\bar{U}_5^5 = (.2)\ (.576) + (.333)\ (.5) + (.467)\ (.56) = .543$

$F_5^5 = \frac{.612 - .543}{1} = .069$

Since F_5^4 is the minimum of F_5^j, j = 1,2,3,4,5, we delete countermeasure CM_4 yielding countermeasure set Y_4 = (1 1 1 0 1) with an index of U_4 = .590 at a cost of E_4 = 26.

Some detailed calculations for the second iteration follow:

Delete CM_1 (CM_4 already deleted)

$\bar{C}_{11.}$ = .1 $\quad C_{1..}$ = (.9) (.9) = .81

$\bar{C}_{12.}$ = .1

$\bar{C}_{21.}$ = .5 $\quad C_{2..}$ = .5

$\bar{C}_{31.}$ = 1.0 $\quad C_{3..}$ = (0) (.7) = 0

$\bar{C}_{32.}$ = .3

$\bar{U}_4^1$ = (.2) (.81) + (.333) (.5) + (.467) (0) = .329

$F_4^1 = \frac{.590 - .329}{10} = .026$

Delete CM_2 (CM_4 already deleted)

$\bar{C}_{11.}$ = .1 $\quad C_{1..}$ = (.9) (0) = 0

$\bar{C}_{12.}$ = 1.0

$\bar{C}_{21.}$ = .5 $\quad C_{2..}$ = .5

$\bar{C}_{31.}$ = .2 $\quad C_{3..}$ = (.8) (0) = 0

$\bar{C}_{32.}$ = 1.0

$\bar{U}_4^2$ = (.2) (0) + (.333) (.5) + (.467) (0) = .167

$F_4^2 = \frac{.590 - .167}{10} = .043$

Delete CM_3 (CM_4 already deleted)

$\bar{C}_{11.}$ = .1 $\quad C_{1..}$ = (.9) (.9) = .81

$\bar{C}_{12.}$ = .1

$\bar{C}_{21.}$ = 1.0 $\quad C_{2..}$ = 0

$\bar{C}_{31.}$ = .2 $\quad C_{3..}$ = .56

$\bar{C}_{32.}$ = .3

$\bar{U}_4^3$ = (.2)(.81) + (.333) (0) + (.467) (.56) = .424

$F_4^3 = \frac{.590 - .424}{5} = .033$

Delete CM_5 (CM_4 already deleted)

$\bar{C}_{11\cdot} = 1.0$ $\qquad$ $C_{1\cdot\cdot} = (0)\ (.9) = 0$

$\bar{C}_{12\cdot} = .1$

$\bar{C}_{21\cdot} = .5$ $\qquad$ $C_{2\cdot\cdot} = .5$

$\bar{C}_{31\cdot} = .2$ $\qquad$ $C_{3\cdot\cdot} = .56$.

$\bar{C}_{32\cdot} = .3$

$\bar{U}_4^5 = (.2)\ (0) + (.333)\ (.5) + (.467)\ (.56) = .428$

$F_4^5 = \dfrac{.590 - .428}{1} = .162$

Since F_4^1 is the minimum of F_4^j, $j = 1,2,3,5$, we delete countermeasure CM_1 yielding countermeasure set $Y_3 = (0\ 1\ 1\ 0\ 1)$ with an index of $U_3 = .329$ at a cost of $E_3 = 16$.

Subsequent iterations delete countermeasures 2,5, and 3 yielding the desired sequence (under $\overline{G2}$) of 3,5,2,1,4 with respective indices of (.167, .167, .329, .590, .612).

The actual computer output for this toy problem is found in Appendix II, Table II-2. (Minor deviations in the figures above and the figures shown in the output are due to round-off errors.)

5. Illustrative Examples - Questions 1 and 2

In this section, we compare the algorithms discussed above in providing answers to questions 1 and 2.

5.1 Generalized Model Example (Many to one correspondence between threats and objectives).

For purposes of illustration, we have considered a hypothetical data base retrieval system. We have assumed our adversary to have four objectives in mind: Objectives O_1, O_2, O_3, O_4. (For purposes of this illustration, we have not clearly defined these objectives.) We have arbitrarily given these objectives relative weights of 8, 7, 5, and 3 yielding normalized weights of 8/23, 7/23, 5/23, and 3/23. In Table 1, we have listed twenty threats against the system and denoted which of the objectives each of the threats accomplish. (The same threat could have appeared for more than one objective - although in this example, they do not.) In Table 2, we have listed 19 countermeasures at our disposal along with the costs of the countermeasures. Table 3 contains hypothetical effectiveness of each of the countermeasures against each of the threats (i.e., the probability that each countermeasure blocks each threat).

We applied algorithms $\overline{G1}$ and $\overline{G2}$ to the above inputs (Tables 1, 2, 3) to determine the sequence of countermeasures to be purchased under each algorithm.

To answer question 2, using algorithm $\overline{G1}$, we would purchase the countermeasures in the order shown in Table 4-1, and to answer question 2 using algorithm $\overline{G2}$ we would purchase the countermeasures in the order shown in Table 4-2.

To answer question 1, we would choose the algorithm and the corresponding countermeasure set which gives the highest security index. For example, if our budget were \$45K, we would use algorithm $\overline{G1}$, yielding an index of .243, using countermeasures 7, 8, 17, 1, 19, and 13 (since

TABLE 1

OBJECTIVES AND THREATS
(Generalized Model Example)

OBJECTIVE 1		Relative weight: 8; Normalized weight: 8/23
Threats:	1	Uncleared user qualifies on a classified accession number
	2	Cleared user's terminal displays classified abstracts that are not related to his work
	3	Inadvertent writing in the direct files or into the pointer table
	4	Uncleared user modifies terminal to transmit the terminal identification of a cleared site
OBJECTIVE 2		Relative weight: 7; Normalized weight: 7/23
Threats	5	User enters illegal strategy for search
	6	Uncleared terminal user falsifies identity by entering the identification of a cleared terminal
	7	User enters an executive statement to perform a function
	8	Unclassified terminal displays a classified document when user guesses or knows the accession number
	9	Information received from a remote terminal is used as an instruction instead of data
	10	Uncleared user accesses the direct file that contains classified abstracts
	11	The check of the restored data bank for accuracy fails
	12	Batch program affects the data transfer operation

OBJECTIVE 3 Relative weight: 5; Normalized weight: 5/23

Threats: 13 Classified abstracts are printed at the central site

14 Illegal command sequences to the system

15 Selected users enter data to be included in the data bases

16 Incorrect direct file pointer

17 Residual information in I/O buffers are displayed to a terminal user

18 Wrong data bank restored from mass storage

OBJECTIVE 4 Relative weight: 3; Normalized weight: 3/23

Threats: 19 Input buffer overflow condition

20 Output buffer overflow condition

TABLE 2

COUNTERMEASURES

No.		Cost
1	Strict formatted entries	5,000
2	Identification from the remote terminal must match the code application program expects to receive from that site	5,000
3	Dedicated phone line/dedicated I/O	1,000
4	Additional logic within each terminal which places terminal identification code to each message it transmits	20,000
5	Input from a terminal is interpreted by the retrieval program	8,000
6	Prior to release of data, user's classification field must be verified	6,000
7	Add logic to check classification indicator in the Direct File pointer	1,000
8	Require authorization prior to transmittal of classified data	15,000
9	Data base access accomplished while off-line	300
10	Staff verifies that the user is allowed to receive the abstract prior to release	300
11	Check fields on document	5,000
12	Separate data bases - one classified and one unclassified	3,000
13	The accession number of the core must match the requested accession number	1,000
14	Direct file is referenced only by the worker segments of the retrieval program (without attempt to access)	4,000
15	File access table must deny the assignment of a classified file to an uncleared user	4,000

No.		Cost
16	Overwrite the output buffer and terminate transmission using an end-of-transmission character check	3,000
17	Requests must be made through the executive program and all checks are made by the executive program	15,000
18	Compare identifier of current user to the identification code in the restored data bank	3,000
19	Polled mode	6,000

TABLE 3

COUNTERMEASURE EFFECTIVENESS MATRIX*

Threats	Countermeasures																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1							.8				.8	.7							
2								.8											
3																	.8		
4	.8	.8																	
5	.8																		
6	.5	.7	.8	.8															
7	.8				.8														
8							.6	.8			.8	.7							
9	.5				.8														
10													.7	.8					
11																	.8		
12																	.8		
13									.8	.8									
14	.5				.8														
15								.8	.8										
16												.7	.7		.7				
17																.8			
18																		.6	
19																			.8
20																			.8

*Entries are Probabilities that Countermeasures Block Threats
Blank Entries are Zero

TABLE 4-1

Algorithm $\overline{G1}$ (Generalized Model Example)

Iteration	Add	Index	Cost ($K)
i	CM No.	Ui	E_i
1	7*	.0	-
2	8*	.0	-
3	17*	.0	-
4	1*	.142	36.0
5	19	.226	42.0
6	13	.243	43.3
7	9**	.243	-
8	18**	.243	-
9	16**	.271	49.3
10	12	.308	52.3
11	2	.356	57.3
12	5	.425	65.3
13	10	.438	65.6
14	3	.449	66.6
15	14	.479	70.6
16	11	.495	75.6
17	15	.500	79.6
18	6	.501	85.6
19	4	.504	105.6

*Add in any order
**Add in any order

TABLE 4-2

Algorithm $\overline{G2}$ (Generalized Model Example)

Iteration	Add	Index	Cost ($K)
i	CM No.	Ui	Ei
1	19	.083	6.0
2	18*	.083	-
3	16*	.083	-
4	13*	.083	-
5	10*	.083	-
6	9*	.083	-
7	1*	.120	18.6
8	12**	.120	-
9	7**	.120	-
10	3**	.120	-
11	17**	.157	38.6
12	8	.332	53.6
13	5	.409	61.6
14	2	.449	66.6
15	14	.479	70.6
16	11	.495	75.6
17	15	.500	79.6
18	4	.503	99.6
19	6	.504	105.6

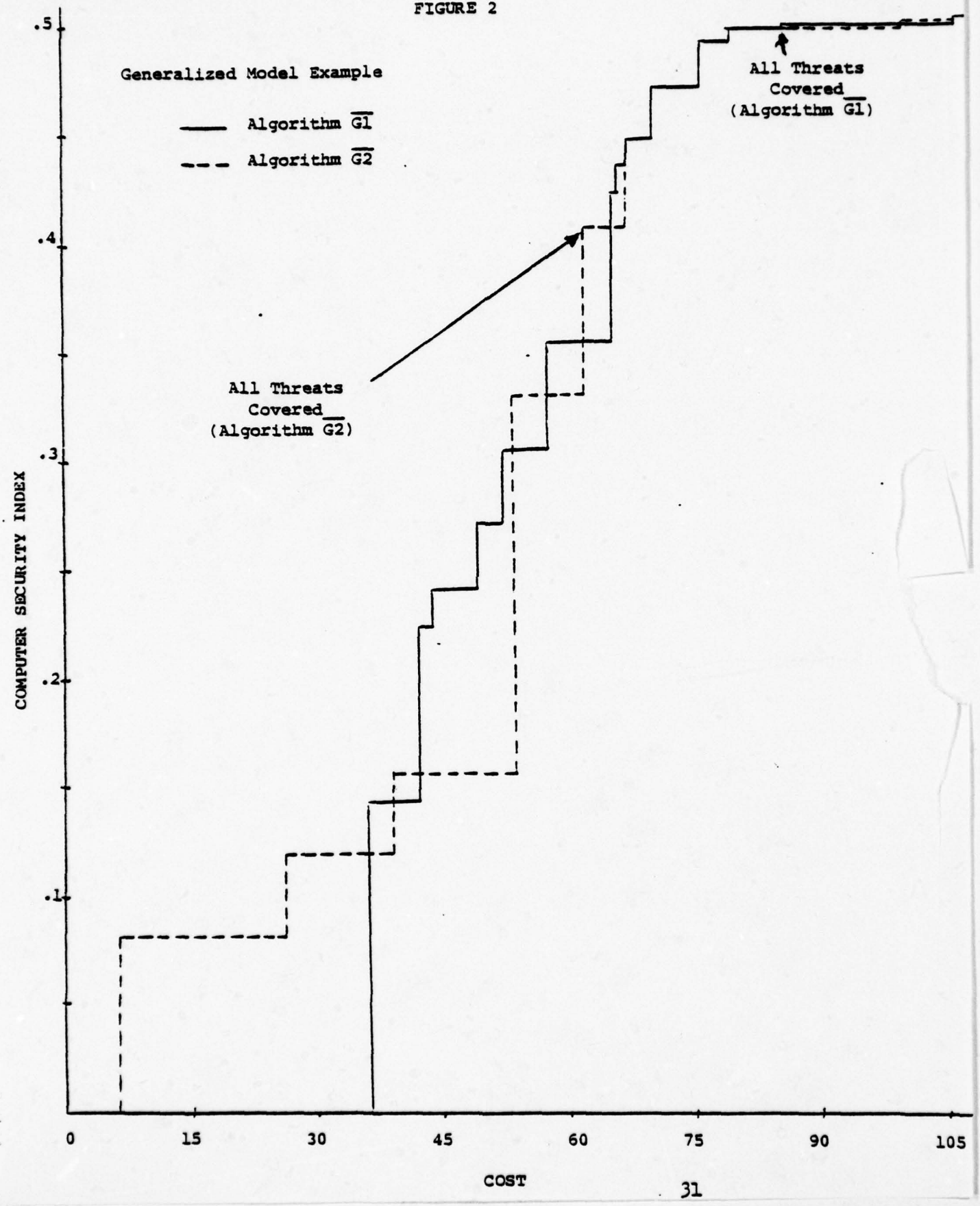
* Add in any order
**Add in any order

for $45K, algorithm $\overline{G2}$ yields an index of only .157). If our budget were $54K, we would use algorithm $\overline{G2}$ yielding an index of .332 using countermeasures 19, 18, 16, 13, 10, 9, 1, 12, 7, 3, 17 and 8 (since for $54K, algorithm $\overline{G1}$ yields an index of only .308). Of course, trade-offs between security index and countermeasure sets could also be considered. For example, a system may have roughly the same security index with two different countermeasure sets which cost roughly the same. Yet one countermeasure set may be smaller, or more easily implemented, or more desirable than the other even though it might have a lower index. In such a case, we might still choose the countermeasure set with the lower index.

A graphical comparison of the two algorithms is contained in Figure 2 (page 31). We see that up to $36K, $\overline{G2}$ is superior. From $36K to $52.3K, $\overline{G1}$ performed better. From $52.3K to $66.6K, $\overline{G1}$ and $\overline{G2}$ alternated and above $66.6K, they were nearly identical. It is interesting to note that under algorithm $\overline{G2}$, all threats have at least some coverage (i.e., nonzero probability of being blocked) for a lower expenditure than under algorithm $\overline{G1}$.

A sample input file for this example is contained in Appendix III in Table III-1. The first entry in the file is the algorithm chosen (G1BAR or G2BAR). (All other input entries are defined in the file.)

FIGURE 2


Generalized Model Example
Algorithm $\overline{G1}$
Algorithm $\overline{G2}$
All Threats Covered (Algorithm $\overline{G1}$)
All Threats Covered (Algorithm $\overline{G2}$)
COMPUTER SECURITY INDEX
.5
.4
.3
.2
.1
0
15
30
45
60
75
90
105
COST

5.2 Simplified Model Example (One-to-one correspondence between threats and countermeasures)

As another illustration, let us consider the simplified model where there is a one-to-one correspondence between threats and objectives. In this case, we must denote a relative weight of each threat which we have done in Table 5. The cost of the countermeasures and the countermeasure effectiveness matrix is assumed to be the same as in the previous example (Table 2 and 3).

Tables 6-1 through 6-4 contain the sequences in which countermeasures should be added for algorithms G1, G2, $\overline{G1}$ and $\overline{G2}$, respectively. As in the case of the generalized model, we make use of these tables to determine the sequence in which countermeasures should be purchased to answer question 2, and the countermeasure set which should be used to answer question 1.

For our sample data, algorithms G1 and $\overline{G1}$ were found to be identical and algorithms G2 and $\overline{G2}$ were nearly identical. Algorithms G2 and $\overline{G2}$ clearly outperformed algorithms G1 and $\overline{G1}$ for expenditures beyond $26.6K. Because of this we would probably choose algorithm G2 or $\overline{G2}$ to answer question 2. For the same reason, we would probably choose algorithms G1 or $\overline{G1}$ to answer question 1 for cost constraints less than $26.6K and choose algorithms G2 or $\overline{G2}$ to answer question 1 for cost constraints more than $26.6K.

As occurred in the previous case, it may be observed that all threats had at least some coverage under algorithms G2 and $\overline{G2}$ (i.e., nonzero probability of being blocked) for a lower expenditure than under algorithms G1 or $\overline{G1}$.

TABLE 5

THREATS
(Simplified Model Example)

Threat No.*	Relative Weight	Normalized Weight
1	8	8/124
2	8	"
3	8	"
4	8	"
5	7	7/124
6	7	"
7	7	"
8	7	"
9	7	"
10	7	"
11	7	"
12	7	"
13	5	5/124
14	5	"
15	5	"
16	5	"
17	5	"
18	5	"
19	3	3/124
20	3	"

*Threat description same as in Table 1

TABLE 6-1

ALGORITHM G1
(Simplified Model Example)

Iteration	Add	Index	Cost ($K)
i	CM No.	U_i	E_i
1	9	.065	.3
2	7	.161	1.3
3	10	.168	1.6
4	13	.235	2.6
5	3	.281	3.6
6	1	.477	8.6
7	16	.509	11.6
8	12	.534	14.6
9	18	.558	17.6
10	14	.572	21.6
11	15	.575	25.6
12	19	.613	31.6
13	2	.628	36.6
14	11	.633	41.6
15	5	.681	49.6
16	17	.823	64.6
17	6	.823	70.6
18	8	.882	85.6
19	4	.883	105.6

TABLE 6-2

ALGORITHM G2
(Simplified Model Example)

Iteration	Add	Index	Cost ($K)
i	CM No.	U_i	E_i
1	9	.065	.3
2	7	.161	1.3
3	13	.230	2.3
4	3	.274	3.3
5	1	.470	8.3
6	10	.477	8.6
7	16	.509	11.6
8	17	.651	26.6
9	12	.676	29.6
10	18	.700	32.6
11	19	.739	38.6
12	5	.787	46.6
13	8	.845	61.6
14	14	.858	65.6
15	2	.873	70.6
16	11	.879	75.6
17	15	.881	79.6
18	4	.882	99.6
19	6	.883	105.6

TABLE 6-3

ALGORITHM $\overline{G1}$
(Simplified Model Example)

Iteration	Add	Index	Cost ($K)
i	CM No.	Ui	Ei
1	9	.065	.3
2	7	.161	1.3
3	10	.168	1.6
4	13	.235	2.6
5	3	.281	3.6
6	1	.477	8.6
7	16	.509	11.6
8	12	.534	14.6
9	18	.558	17.6
10	14	.572	21.6
11	15	.575	25.6
12	19	.613	31.6
13	2	.628	36.6
14	11	.633	41.6
15	5	.681	49.6
16	17	.823	64.6
17	6	.823	70.6
18	8	.882	85.6
19	4	.883	105.6

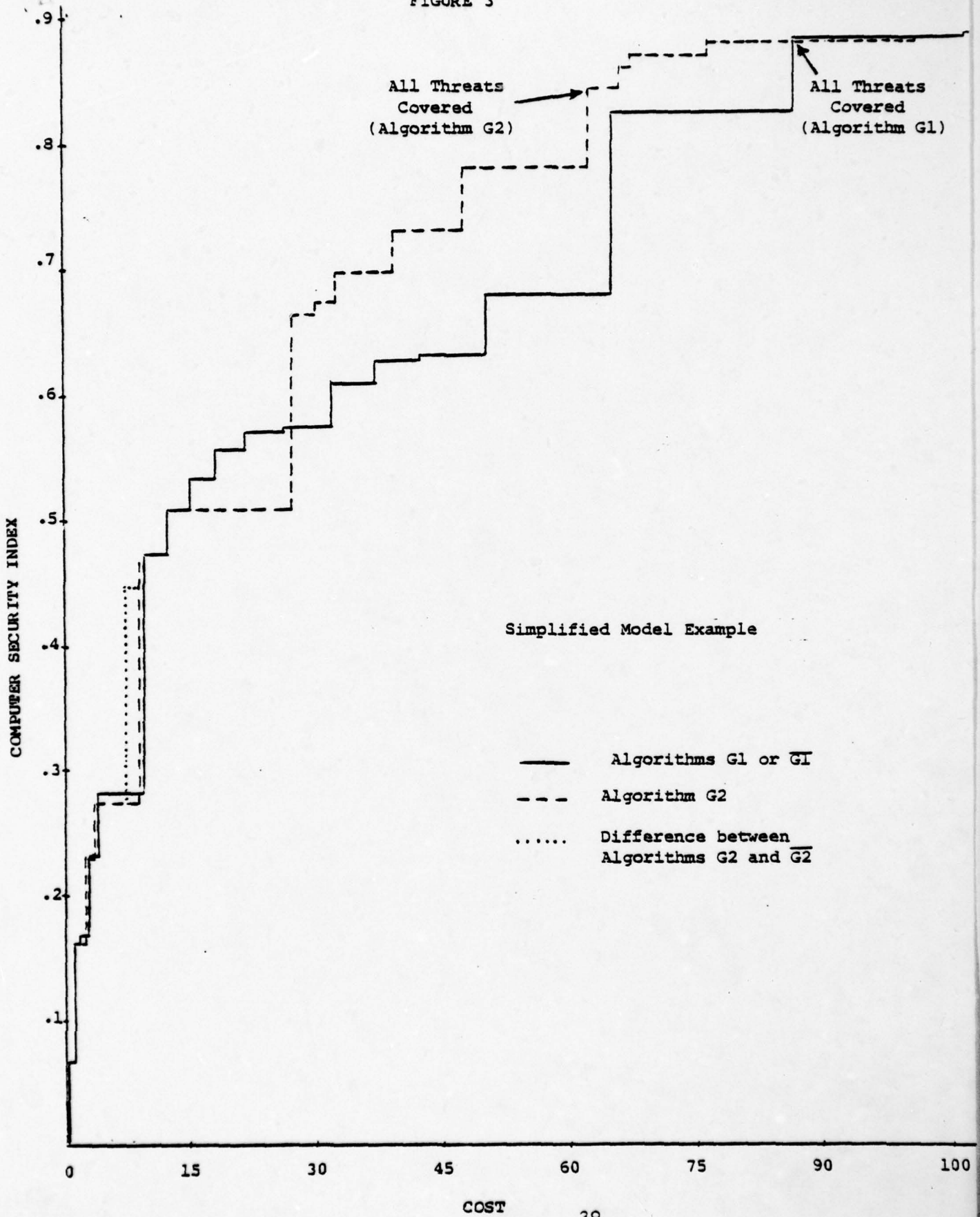
TABLE 6-4

ALGORITHM $\overline{G2}$
(Simplified Model Example)

Iteration	Add	Index	Cost ($K)
i	CM No.	Ui	Ei
1	9	.065	.3
2	7	.161	1.3
3	13	.229	2.3
4	1	.448	7.3
5	3	.470	8.3
6	10	.476	8.6
7	16	.509	11.6
8	17	.651	26.6
9	12	.676	29.6
10	18	.700	32.6
11	19	.739	38.6
12	5	.787	46.6
13	8	.845	61.6
14	14	.858	65.6
15	2	.873	70.6
16	11	.879	75.6
17	15	.881	79.6
18	4	.882	99.6
19	6	.883	105.6

A graphical comparison of the four algorithms is contained in Figure 3 (page 39). A sample input file for the example is contained in Appendix III, Table III-2.

FIGURE 3


All Threats Covered (Algorithm G2)
All Threats Covered (Algorithm G1)
COMPUTER SECURITY INDEX
.9
.8
.7
.6
.5
.4
.3
.2
.1
0
15
30
45
60
75
90
100
COST
Simplified Model Example
Algorithms G1 or $\overline{G1}$
Algorithm G2
Difference between Algorithms G2 and $\overline{G2}$

5.3 Some Comments

Whether the results we have observed in Figures 2 and 3 for our hypothetical examples hold true in general has not been studied in this effort. However, algorithms G2 and $\overline{G2}$, which performed as well or better than algorithms G1 and $\overline{G1}$, appear more intuitively appealing since they add countermeasures to the countermeasure set so that the increase in computer security per dollar is maximized.

As mentioned earlier, we recommend running all four algorithms for the simplified model (one-to-one correspondence between threats and objectives) and algorithms $\overline{G1}$ and $\overline{G2}$ for the generalized model (many-to-one correspondence between threats and objectives).

6. Illustrative Example - Question 3

We have no illustrative example for Question 3. No effort has been made in this paper to develop a standard set of threats and countermeasures. If a standard set of threats and countermeasures were available for all systems being considered, we would calculate the computer security index for each system and choose the one with the highest index for the given budget constraint. However, trade-offs between security and cost might still be considered. For example, one system may have nearly as high security index as another but at a much lower cost so that we still might choose the system with the lower index.

If it is not already obvious, we might point out that if a particular threat is not applicable to a specific system, it may be entered but negated by assigning it a zero weight; and a "built-in" countermeasure may be handled by assigning it zero cost.

7. Illustrative Example - Question 4.

For our example, let us use the simplified model example of Section 5.2, i.e., the case where is a one to one correspondence between threats and countermeasures (using Tables 3, 2, and 5 as our input).

Let us arbritarily assume that one package contains the first ten countermeasures and the other package contains the last nine countermeasures.

A sample input file for the last nine counteremeasure set is in Appendix III - Table III-3, where our first entry is "CSI ONLY" indicating we want only the computer security index calculated.* (All input entries are described in the file.)

The result of running the program (once for each countermeasure set) is that the first ten countermeasure set has an index of .537 at a cost of $61.6K and the last nine countermeasure set has an index of .443 at a cost of $44K. If one were deciding between these two countermeasure sets, one would have to be determine if the additional index (.537 as opposed to .443) is worth the additional cost ($61.6K as opposed to $44K).

---

*NOTE: If the input file were already set up for all 19 countermeasures and the index for the first 10 countermeasures were desired, it may be more convenient to use the "PRESET" option (to be discussed in the next section) (rather than the "CSI only" option) and input the sequence so that the first 10 counteremasures appeared in the beginning the sequence. The desired index would then appear at iteration 10 of the output.

8. Illustrative Example - Question 5.

For our example, let us use the generalized model example of Section 5.1 where we use Tables 1, 2, and 3 as our inputs. Let us assume we are interested in plotting the computer security index versus cost for the following (arbitrarily selected) sequence of countermeasures: 4, 17, 8, 5, 19, 6, 11, 2, 1, 15, 14, 18, 16, 12, 13, 7, 3, 10, 9. A sample input file is in Appendix III, Table III-4, where our first input entry is PRESET and where we subsequently enter the above sequence. (All input entries are described in the file.) The output is similar in nature to the output file shown in Appendix II, Table II - 2, except that the sequence outputted would be the sequence above. Actual results of the output is contained in Table 7. If we wanted to compare this sequence with another sequence, we would run the program again with the new sequence substituted for the above sequence. The results could then be compared and an appropriate decision made as to which sequence to implement.

TABLE 7

PRE-SELECTED SEQUENCE
(Generalized Model Example)

Iteration	Add	Index	Cost ($K)
i	CM No.	Ui	Ei
1	4	.0	20.0
2	17	.0	35.0
3	8	.0	50.0
4	5	.0	58.0
5	19	.083	64.0
6	6	.083	70.0
7	11	.083	75.0
8	2	.223	80.0
9	1	.254	85.0
10	15	.254	89.0
11	14	.351	93.0
12	18	.351	96.0
13	16	.351	99.0
14	12	.386	102.0
15	13	.404	103.0
16	7	.417	104.0
17	3	.420	105.0
18	10	.478	105.3
19	9	.504	105.6

9. Description of the Computer Program

A computer program has been written in FORTRAN for the CDC-7600 both to calculate the computer security index for any given countermeasure set and to generate the sequence of countermeasures called for by algorithms G1, G2, $\bar{G}1$, and $\bar{G}2$. The program has already been referenced many times in the course of this paper. A listing of the program is found in Appendix 1.

Appendix 1 also contains a list and description of all variables used in the program. In usihg the program, one should insure that the variables are properly dimensioned for the problem being considered. (See Appendix 1 for guidelines in this matter).

To indicate the general nature of the input and output files, sample input and output files for the running of the toy problem of Section 4 has been included in Appendix II.

10. Acknowledgement

The work in this paper was a follow-on refinement and generalization of work performed by Mr. A. B. Marsh, III.

In Mr. Marsh's original work, he considered the special case of the model discussed above (i.e., the case where there is a one to one correspondence between threats and countermeasures) and proposed an algorithm (algorithm G1 above) for determining the sequence in which the countermeasures should be purchased. His suggestions for further study helped motivate the current study.

List of Tables

APPENDIX I

This appendix contains a listing of the computer program and definitions of the variables used in the program.

When running the program one should insure that the variables are properly dimensioned for the problem being considered. To do this, refer to the REAL and INTEGER dimension statements of the Program Listing which has the following correspondence between dimension numbers and the quantities they represent:

20 - Number of threats

19 - Number of countermeasures

8 - Maximum number of threats listed for any objective.

PROGRAM LISTING

```
      PROGRAM CSI
      REAL ABSWT(20),SUMWT,WEIGHT(20),INDEX(19),
     1 MATRIX(20,8,19),INDEXP,F(20),TEMP,TEMP1,
     2 FACTOR(20,8),THR(20,8),FACTV(20,8),
     3 THRV(20),COST(19),COSTP,INDEXP1,FACT(19),
     4 TEMP2
      INTEGER N,T,FLAG1,CM(19),M(20),DELETE,MK,
     1 ITER,ALG,OPT1,SEQ(19),FLAG2
      READ (5,249)
  249 FORMAT (/////////////////)
      READ (5,230) ALG
  230 FORMAT (R5)
      IF((ALG.EQ.5RG1    ) .OR. (ALG.EQ.5RG2    ))
     1    OPT1=1
      IF((ALG.EQ.5RG1BAR) .OR. (ALG.EQ.5RG2BAR))
     1    OPT1=2
      IF(ALG.EQ.5RPRESE) OPT1=3
      IF(ALG.EQ.5RCSI 0) OPT1=4
      READ (5,200) T,N
  200 FORMAT (10I5)
      READ(5,240)
  240 FORMAT(////)
      IF(OPT1.EQ.3) READ(5,241) (SEQ(I),I=1,N)
  241 FORMAT(10I5)
      WRITE(6,901) T
  901 FORMAT(22HNUMBER OF OBJECTIVES =,I3)
      WRITE(6,903)
  903 FORMAT(32HNUMBER OF THREATS PER OBJECTIVE:)
      READ (5,200) (M(K),K=1,T)
      WRITE (6,902) (M(K),K=1,T)
  902 FORMAT(7I7,1X)
      READ (5,203) (ABSWT(K), K=1,T)
  203 FORMAT (10F5.0)
      WRITE (6,905)
  905 FORMAT(28HRELATIVE WEIGHTS OF THREATS:)
      WRITE (6,904) (ABSWT(K), K=1,T)
  904 FORMAT(7F7.2,1X)
      SUMWT = 0.0
      DO 501 K=1,T
  501 SUMWT = SUMWT + ABSWT(K)
      DO 502 K=1,T
  502 WEIGHT(K) = ABSWT(K)/SUMWT
      WRITE(6,910)
  910 FORMAT(30HNORMALIZED WEIGHTS OF THREATS:)
      WRITE (6,800) (WEIGHT(K), K=1,T)
```

```
  800 FORMAT (F10.5)
      WRITE (6,906) N
  906 FORMAT(27HNUMBER OF COUNTERMEASURES =,I3)
      WRITE(6,907)
  907 FORMAT(24HCOST OF COUNTERMEASURES:)
      READ (5,202) (COST(J), J=1,N)
  202 FORMAT (5F10.0)
      WRITE(6,908) (COST(J),J=1,N).
  908 FORMAT(5F10.0)
      COSTP = 0.0
      DO 590 J=1,N
      COSTP = COSTP + COST(J)
  590 CONTINUE
      WRITE(6,912)
  912 FORMAT(
     140HCOUNTERMEASURE EFFECTIVENESS MATRIX:     )
      DO 504 K=1,T
      MK = M(K)
      DO 5041 I=1,MK
      READ (5,201) (MATRIX(K,I,J), J=1,N)
  201 FORMAT (10F5.1)
      WRITE (6,911) (MATRIX(K,I,J), J=1,N)
  911 FORMAT(5F10.2)
      WRITE(6,921)
  921 FORMAT(/)
 5041 CONTINUE
  504 CONTINUE.
      DO 701 K=1,T
      MK = M(K)
      DO 701 I=1,MK
      DO 7011 J=1,N
 7011 MATRIX (K,I,J) = 1.0 - MATRIX(K,I,J)
  701 CONTINUE
      INDEXP = 0.0
      DO 505 K=1,T
      F(K) = 1.0
      MK = M(K)
      DO 5051 I=1,MK
      TEMP = 1.0
      DO 5052 J=1,N
 5052 TEMP = TEMP*MATRIX(K,I,J)
      FACTOR(K,I) = TEMP
      TEMP = 1.0 - TEMP
      THR(K,I) = TEMP
      F(K) = TEMP * F(K)
 5051 CONTINUE
```

```
      INDEXP = WEIGHT(K)*F(K) + INDEXP
  505 CONTINUE
      INDEXP1=INDEXP
      WRITE (6,915) INDEXP
  915 FORMAT(
     140HCOMPUTER SECURITY INDEX FOR FULL          /
     238HCOUNTERMEASURE SET ENTERED IS EQUAL TO,
     3                    F10.5)
      WRITE(6,916) COSTP
  916 FORMAT(
     140HTHE COST FOR THE FULL COUNTERMEASURE SET/
     219HENTERED IS EQUAL TO, F10.0)
      WRITE(6,925)
  925 FORMAT(///)
      IF(OPT1.EQ.4) GO TO 99
      WRITE(6,926)
  926 FORMAT(
     140H**********************************************)
      WRITE(6,925)
      IF(OPT1.EQ.3) WRITE(6,9271)
 9271 FORMAT(
     140HTHE SEQUENCE OF COUNTERMEASURES SELECTED/
     240HBELOW HAS BEEN PRESELECTED BY THE PERSON/
     340HCURRENTLY RUNNING THE PROGRAM AND DOES   /
     440HNOT NECESSARILY REFLECT THE OUTCOME OF   /
     540HTHE ALGORITHMS G1,G2,G1BAR,OR G2BAR      )
      IF(OPT1.NE.3) WRITE(6,927) ALG
  927 FORMAT(16HSTART ALGORITHM ,R5)
      WRITE(6,925)
  810 FORMAT (5F10.5)
      IF(OPT1 .EQ. 2) CALL ERASER (CM,N,1)
      IF(OPT1 .NE. 2) CALL ERASER (FACTOR,200,1.0)
      ITER = 0
      IF(OPT1 .NE. 2) INDEXP = 0.0
      IF(OPT1 .NE. 2) TEMP2=0.0
      IF(OPT1 .NE. 2) COSTP = 0.0
  560 CONTINUE
      ITER = ITER + 1
      IF(ITER .EQ. N+1) GO TO 561
      IF(OPT1.NE.2) TEMP1=0.0
      IF(OPT1.EQ.2) FLAG2=1
      IF(OPT1.EQ.3) GO TO 577
      WRITE (6,917) ITER
  917 FORMAT(
     140HBELOW, IF COUNTERMEASURE X WERE           /
     226HADDED(DROPPED) AT ITERATION ,I3,1H,       /
```

```
     340H THEN THE VALUE OF THE FACTOR TO BE       /
     440HMAXIMIZED (MINIMIZED) IS Y AND THE NEW    /
     540HINDEX, IF X WERE ADDED (DROPPED)          /
     640HWOULD BE Z                                )
      WRITE(6,919)
 919 FORMAT(2X,1HX,6X,1HY,10X,1HZ)
 577 CONTINUE
      DO 530  J=1,N
      IF(OPT1.EQ.3 .AND. J.NE.SEQ(ITER)) GO TO 530
      INDEX(J) = 0.0
      IF (CM(J).EQ.0 .AND. OPT1.EQ.2) GO TO 530
      IF (CM(J).EQ.1 .AND. OPT1.NE.2) GO TO 530
      DO 5302 K=1,T
      MK = M(K)
      DO 5302 I=1,MK
      IF(MATRIX(K,I,J).EQ.1.0 .AND. OPT1.EQ.2)
     1                                  GO TO 601
      IF(OPT1.EQ.2)
     1FACTV(K,I) = 1.0 - FACTOR(K,I)/MATRIX(K,I,J)
      IF(OPT1.NE.2)
     1FACTV(K,I) = 1.0 - FACTOR(K,I)*MATRIX(K,I,J)
      GO TO 5302
 601  FACTV(K,I) = 1.0 - FACTOR(K,I)
5302  CONTINUE
      DO 400 K=1,T
      MK = M(K)
 400  CONTINUE
      DO 5303 K=1,T
      THRV(K) = 1.0
      MK=M(K)
      DO 5303 I=1,MK
5303  THRV(K) = THRV(K) * FACTV(K,I)
      DO 5304 K=1,T
5304  INDEX(J) = WEIGHT(K)*THRV(K) + INDEX(J)
      IF(OPT1.EQ.3 .AND. J.EQ.SEQ(ITER)) GO TO 573
      IF((INDEX(J).LE.0.00000001).AND.(OPT1.EQ.2))
     1                                    GO TO 760
      IF(ALG .EQ. SRG1    )
     1       FACT(J)=INDEX(J)/(COSTP+COST(J))
      IF(ALG .EQ. SRG2    )
     1       FACT(J)=(INDEX(J)-INDEXP)/COST(J)
      IF(ALG .EQ. SRG1BAR)
     1       FACT(J)=(COSTP-COST(J))/INDEX(J)
      IF(ALG .EQ. SRG2BAR)
     1       FACT(J)=(INDEXP-INDEX(J))/COST(J)
 760  CONTINUE
```

```
      IF((INDEX(J).LE.0.00000001).AND.(OPT1.EQ.2))
     1                          GO TO 5291
      GO TO 5292
 5291 INDEX(J)=0.0
      IF(ALG.EQ.SRG2BAR) GO TO 5292
      WRITE(6,8251) J,INDEX(J)
 8251 FORMAT(I3,4X,5HLARGE,1X,F10.5)
      GO TO 530
 5292 CONTINUE
      IF(OPT1.NE.3)
     1            WRITE (6,825) J,FACT(J),INDEX(J)
  825 FORMAT(I3,E10.1,F10.5)
      IF(FLAG2.EQ.1 .AND. OPT1.EQ.2) GO TO 5003
      IF(FACT(J).GE.TEMP1 .AND. OPT1.EQ.2)
     1                                GO TO 530
      IF(FACT(J).LE.TEMP1 .AND. OPT1.EQ.1)
     1                                GO TO 530
 5003 CONTINUE
      FLAG2=0
      DELETE = J
      TEMP1 = FACT(J)
  530 CONTINUE
  573 CONTINUE
      IF(OPT1.EQ.2) CM(DELETE) = 0
      IF(OPT1.EQ.3) DELETE=SEQ(ITER)
      IF(OPT1.NE.2) CM(DELETE) = 1
      INDEXP = INDEX(DELETE)
      IF((INDEXP.LE.0.00000001) .AND. (OPT1.EQ.2))
     1                              GO TO 561
      IF(ITEMP3 .EQ.DELETE) FLAG1=1
      IF(TEMP2.EQ.INDEXP .AND. OPT1.EQ.1)
     1                             FLAG1=1
      IF(FLAG1 .EQ. 1) WRITE(6,8216)
 8216 FORMAT(////
     140HATTENTION  -----  STOP                          /
     240HADDITION OF ANY COUNTERMEASURE DOES NOT        /
     340HINCREASE INDEX.                                /
     440HYOU SHOULD BE USING ALGORITHM G1BAR OR         /
     540HALGORITHM G2BAR                                )
      IF(FLAG1 .EQ. 1) GO TO 99
      TEMP2=INDEXP
      ITEMP3=DELETE
      IF(OPT1.EQ.2) COSTP=COSTP-COST(DELETE)
      IF(OPT1.NE.2) COSTP=COSTP+COST(DELETE)
```

```
      DO 580 K=1,T
      MK = M(K)
      DO 580 I=1,MK
      IF(OPT1 .EQ. 2)
     1FACTOR(K,I) = FACTOR(K,I)/MATRIX(K,I,DELETE)
      IF(OPT1 .NE. 2)
     1FACTOR(K,I) = FACTOR(K,I)*MATRIX(K,I,DELETE)
  580 CONTINUE
      WRITE (6,8211) ITER
 8211 FORMAT (9HITERATION.I3)
      IF (OPT1.EQ.2) WRITE (6,8212) DELETE
 8212 FORMAT (6HDELETE, I3)
      IF (OPT1.NE.2) WRITE (6,8214) DELETE
 8214 FORMAT (3HADD,I3)
      WRITE (6,8213) (CM(J), J=1,N)
 8213 FORMAT (6HCM SET, 20I2)
      WRITE (6,900) INDEXP,COSTP
  900 FORMAT (5HINDEX,F10.5,5X,4HCOST,F10.0///)
      IF(INDEXP.EQ.INDEXP1 .AND. OPT1.NE.2)
     1                                    GO TO 561
      GO TO 560
  561 CONTINUE
      IF(OPT1.NE.2) GO TO 99
      WRITE(6,921)
      WRITE (6,2000)
 2000 FORMAT(
     140HINDEX WILL DROP TO ZERO                    /
     240HWITH FURTHER DELETIONS.                    )
   99 CONTINUE
      WRITE(6,925)
      STOP
      END
%     END
```

DEFINITION OF VARIABLES USED IN COMPUTER PROGRAM

Text Symbol (if applicable) (See Section 3)	Program Variable	Definition
w_k	ABSWT(k)	Relative weight of objective k
-	ALG	Designates options for running programs (also see OPT1)
x_j	CM(j)	Indicates whether countermeasure j is in countermeasure set
D_j	COST(j)	Cost of countermeasure j
-	COSTP	Sum of costs of countermeasure sets
-	DELETE	Countermeasure number added or deleted
C_k..	F(k)	Probability that all threats against objective k are blocked (for full countermeasure set entered)
-	FACT(j)	Quantity to be maximized (or minimized) during operation of algorithms
$\overline{C}_{ki}$.	FACTOR(k,i)	Probability that the i^{th} threat of objective k is not blocked
C_{ki}.	FACTV(k,i)	Probability that the i^{th} threat against objective k is blocked
-	FLAG1	Dummy variable which sets to one if index does not increase as countermeasures are added under algorithms G1 and G2
-	FLAG2	Dummy variable which is used in operation of algorithms $\overline{G1}$ and $\overline{G2}$
U_1^j ($\overline{U}_1^j$)-	INDEX(j)	Computer security index if j^{th} countermeasure is added (or dropped)
-	INDEXP	Computer security index

Text Symbol (if applicable) (See Section 3)	Program Variable	Definition
-	ITEMP3	Temporary storage location for DELETE
-	ITER	Iteration number in executing algorithms
I_k	M(k)	Number of threats for objective k
Ckij	MATRIX(k,i,j)	Probability that j^{th} countermeasure blocks i^{th} threat of objective k (later changed to (1-probability)
I_k	MK	Same as M(k)
n	N	Number of countermeasures
-	OPT1	Designates options for running programs 1. Algorithms G1 or G2 implemented 2. Algorithms $\overline{G1}$ or $\overline{G2}$ implemented 3. Computer security index calculated for pre-selected sequence of countermeasures 4. Computer security index calculated-algorithms are not run
-	SEQ(j)	Sequence of countermeasures to be examined if user submits own sequence
$\sum_{t=1}^{K} w_t$	SUMWT	Summation of ABSWT(k)
K	T	Number of objectives
-	TEMP1	Temporary storage location for FACT(j)
-	TEMP2	Temporary storage location for computer security index
Cki·	THR(k,i)	Probability that the i^{th} threat against objective k is blocked (for full countermeasure set entered)
Ck··	THRV(k)	Probability that all threats against objective k are blocked
Wk	WEIGHT(k)	Normalized weight of objective k

APPENDIX II

This appendix contains a sample input file and a sample output file of the computer program. The two sample files chosen are for the toy problem of Section 4.

TABLE II-1
SAMPLE INPUT FILE
(TOY PROBLEM)

```
DATA SET FOR PROGRAM CSI
ENTER HERE:
PARAMETERS      REPRESENTING                    FORMAT
T,N             NO. OF OBJECTIVES               2I5
                NO. OF COUNTERMEASURES
M(K)            NO. OF THREATS PER              (10I5)
                OBJECTIVE K
ABSWT(K)        REL. WT. OF OBJECT. K           (10I5)
COST(J)         COST OF COUNTERMEASURE J        (5I10)
MATRIX(K,I,J) COUNTERMEASURE MATRIX             (10F5.1)
BEFORE ENTERING ANY OF THE PARAMETERS ABOVE, ENTER
ON THE NEXT LINE THE ALGORITHM DESIRED (E.G. G1,
G2,G1BAR,G2BAR).  IF YOU ARE USING A PRE-SELECTED
SEQUENCE, ENTER *PRESET* ON THE NEXT LINE.  IF YOU
WANT ONLY THE COMPUTER SECURITY INDEX
CALCULATED, ENTER *CSI ONLY* ON THE NEXT LINE
G2BAR
    3    5
(IF YOU ARE USING A PRE-SELECTED SEQUENCE OF
COUNTERMEASURES, I.E. YOU HAVE ENTERED PRESET
ABOVE, THEN ENTER THAT SEQUENCE ON THE NEXT LINE
IN FORMAT (10I5).  OTHERWISE, CONTINUE ENTERING
PARAMETERS AS DIRECTED ABOVE.)
    2    1    2
    3    5    7
         10        10        5         3         1
   .0   .0   .0   .6   .9
   .0   .9   .0   .6   .0
   .0   .0   .5   .0   .0
   .8   .0   .0   .0   .0
   .0   .7   .0   .0   .0


*EOF*
```

TABLE II-2
SAMPLE OUTPUT FILE
(TOY PROBLEM)

```
NUMBER OF OBJECTIVES =  3
NUMBER OF THREATS PER OBJECTIVE:
     2       1       2
RELATIVE WEIGHTS OF THREATS:
   3.00    5.00    7.00
NORMALIZED WEIGHTS OF THREATS:
    .20000
    .33333
    .46667
NUMBER OF COUNTERMEASURES =  5
COST OF COUNTERMEASURES:
      10          10          5           3           1
COUNTERMEASURE EFFECTIVENESS MATRIX:
     0.00        0.00        0.00         .60         .90

     0.00         .90        0.00         .60        0.00

     0.00        0.00         .50        0.00        0.00

      .80        0.00        0.00        0.00        0.00

     0.00         .70        0.00        0.00        0.00

COMPUTER SECURITY INDEX FOR FULL
COUNTERMEASURE SET ENTERED IS EQUAL TO     .61232
THE COST FOR THE FULL COUNTERMEASURE SET
ENTERED IS EQUAL TO         29

**********************************************

START ALGORITHM G2BAR
```

TABLE II-2 (CONTINUED)

```
BELOW, IF COUNTERMEASURE X WERE
ADDED(DROPPED) AT ITERATION   1,
THEN THE VALUE OF THE FACTOR TO BE
MAXIMIZED (MINIMIZED) IS Y AND THE NEW
INDEX, IF X WERE ADDED (DROPPED)
WOULD BE Z
  X       Y          Z
  1    2.6E-02     .35099
  2    3.3E-02     .28187
  3    3.3E-02     .44565
  4    7.4E-03     .59000
  5    6.9E-02     .54320
ITERATION  1
DELETE  4
CM SET 1 1 1 0 1
INDEX     .59000      COST         26


BELOW, IF COUNTERMEASURE X WERE
ADDED(DROPPED) AT ITERATION   2,
THEN THE VALUE OF THE FACTOR TO BE
MAXIMIZED (MINIMIZED) IS Y AND THE NEW
INDEX, IF X WERE ADDED (DROPPED)
WOULD BE Z
  X       Y          Z
  1    2.6E-02     .32867
  2    4.2E-02     .16667
  3    3.3E-02     .42333
  5    1.6E-01     .42800
ITERATION  2
DELETE  1
CM SET 0 1 1 0 1
INDEX     .32867      COST         16


BELOW, IF COUNTERMEASURE X WERE
ADDED(DROPPED) AT ITERATION   3,
THEN THE VALUE OF THE FACTOR TO BE
MAXIMIZED (MINIMIZED) IS Y AND THE NEW
INDEX, IF X WERE ADDED (DROPPED)
WOULD BE Z
  X       Y          Z
  2    1.6E-02     .16667
  3    3.3E-02     .16200
  5    1.6E-01     .16667
ITERATION  3
DELETE  2
CM SET 0 0 1 0 1
INDEX     .16667      COST          6
```

TABLE II-2 (CONTINUED)

```
BELOW, IF COUNTERMEASURE X WERE
ADDED(DROPPED) AT ITERATION   4.
THEN THE VALUE OF THE FACTOR TO BE
MAXIMIZED (MINIMIZED) IS Y AND THE NEW
INDEX, IF X WERE ADDED (DROPPED)
WOULD BE Z
   X         Y            Z
   3    3.3E-02     0.00000
   5    8.9E-16      .16667
ITERATION  4
DELETE  5
CM SET 0 0 1 0 0
INDEX      .16667      COST          5


BELOW, IF COUNTERMEASURE X WERE
ADDED(DROPPED) AT ITERATION   5.
THEN THE VALUE OF THE FACTOR TO BE
MAXIMIZED (MINIMIZED) IS Y AND THE NEW
INDEX, IF X WERE ADDED (DROPPED)
WOULD BE Z
   X         Y            Z
   3    3.3E-02     0.00000


INDEX WILL DROP TO ZERO
WITH FURTHER DELETIONS.


*EOF*
```

APPENDIX III

This appendix contains sample input files for examples considered in other sections of the text.

Sample Output File	Relevant Section
Table III-1	5.1
Table III-2	5.2
Table III-3	7
Table III-4	8

TABLE III-1
SAMPLE INPUT FILE
(GENERALIZED MODEL EXAMPLE)

```
DATA SET FOR PROGRAM CSI
ENTER HERE:
PARAMETERS      REPRESENTING                    FORMAT
T,N             NO. OF OBJECTIVES               2I5
                NO. OF COUNTERMEASURES
M(K)            NO. OF THREATS PER              (10I5)
                OBJECTIVE K
ABSWT(K)        REL. WT. OF OBJECT. K           (10I5)
COST(J)         COST OF COUNTERMEASURE J        (5I10)
MATRIX(K,I,J) COUNTERMEASURE MATRIX             (10F5.1)
BEFORE ENTERING ANY OF THE PARAMETERS ABOVE, ENTER
ON THE NEXT LINE THE ALGORITHM DESIRED (E.G. G1,
G2,G1BAR,G2BAR). IF YOU ARE USING A PRE-SELECTED
SEQUENCE, ENTER *PRESET* ON THE NEXT LINE. IF YOU
WANT ONLY THE COMPUTER SECURITY INDEX
CALCULATED, ENTER *CSI ONLY* ON THE NEXT LINE
G2BAR
    4   19
(IF YOU ARE USING A PRE-SELECTED SEQUENCE OF
COUNTERMEASURES, I.E. YOU HAVE ENTERED PRESET
ABOVE, THEN ENTER THAT SEQUENCE ON THE NEXT LINE
IN FORMAT (10I5). OTHERWISE, CONTINUE ENTERING
PARAMETERS AS DIRECTED ABOVE.)
    4    8    6    2
    8    7    5    3
      5000      5000      1000     20000      8000
      6000      1000     15000       300       300
      5000      3000      1000      4000      4000
      3000     15000      3000      6000
  .0   .0   .0   .0   .0   .0   .8   .0   .0   .0
  .8   .7   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .8   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .8   .0   .0
  .8   .8   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0
  .8   .0   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0
```

REMAINDER OF COUNTERMEASURE EFFECTIVENESS DATA
HAS BEEN OMITTED

TABLE III-2
SAMPLE INPUT FILE
(SIMPLIFIED MODEL EXAMPLE)

```
DATA SET FOR PROGRAM CSI
ENTER HERE:
PARAMETERS      REPRESENTING                        FORMAT
T,N             NO. OF OBJECTIVES                   2I5
                NO. OF COUNTERMEASURES
M(K)            NO. OF THREATS PER                  (10I5)
                OBJECTIVE K
ABSWT(K)        REL. WT. OF OBJECT. K               (10I5)
COST(J)         COST OF COUNTERMEASURE J            (5I10)
MATRIX(K,I,J)   COUNTERMEASURE MATRIX              (10F5.1)
BEFORE ENTERING ANY OF THE PARAMETERS ABOVE, ENTER
ON THE NEXT LINE THE ALGORITHM DESIRED (E.G. G1,
G2,G1BAR,G2BAR).  IF YOU ARE USING A PRE-SELECTED
SEQUENCE, ENTER *PRESET* ON THE NEXT LINE.  IF YOU
WANT ONLY THE COMPUTER SECURITY INDEX
CALCULATED, ENTER *CSI ONLY* ON THE NEXT LINE
G2BAR
  20  19
(IF YOU ARE USING A PRE-SELECTED SEQUENCE OF
COUNTERMEASURES, I.E. YOU HAVE ENTERED PRESET
ABOVE, THEN ENTER THAT SEQUENCE ON THE NEXT LINE
IN FORMAT (10I5).  OTHERWISE, CONTINUE ENTERING
PARAMETERS AS DIRECTED ABOVE.)
    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1
    8    8    8    8    7    7    7    7    7    7
    7    7    5    5    5    5    5    5    3    3
      5000      5000      1000     20000      8000
      6000      1000     15000       300       300
      5000      3000      1000      4000      4000
      3000     15000      3000      6000
  .0   .0   .0   .0   .0   .0   .8   .0   .0   .0
  .8   .7   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .8   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .8   .0   .0
  .8   .8   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0
  .8   .0   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0
```

REMAINDER OF COUNTERMEASURE EFFECTIVENESS DATA
HAS BEEN OMITTED

TABLE III-3
SAMPLE INPUT FILE
(SIMPLIFIED MODEL EXAMPLE)

```
DATA SET FOR PROGRAM CSI
ENTER HERE:
PARAMETERS      REPRESENTING                        FORMAT
T,N             NO. OF OBJECTIVES                   2I5
                NO. OF COUNTERMEASURES
M(K)            NO. OF THREATS PER                  (10I5)
                OBJECTIVE K
ABSWT(K)        REL. WT. OF OBJECT. K               (10I5)
COST(J)         COST OF COUNTERMEASURE J            (5I10)
MATRIX(K,I,J)   COUNTERMEASURE MATRIX              (10F5.1)
BEFORE ENTERING ANY OF THE PARAMETERS ABOVE, ENTER
ON THE NEXT LINE THE ALGORITHM DESIRED (E.G. G1,
G2,G1BAR,G2BAR).  IF YOU ARE USING A PRE-SELECTED
SEQUENCE, ENTER *PRESET* ON THE NEXT LINE.  IF YOU
WANT ONLY THE COMPUTER SECURITY INDEX
CALCULATED, ENTER *CSI ONLY* ON THE NEXT LINE
CSI ONLY
    4    9
(IF YOU ARE USING A PRE-SELECTED SEQUENCE OF
COUNTERMEASURES, I.E. YOU HAVE ENTERED PRESET
ABOVE, THEN ENTER THAT SEQUENCE ON THE NEXT LINE
IN FORMAT (10I5).  OTHERWISE, CONTINUE ENTERING
PARAMETERS AS DIRECTED ABOVE.)
    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1
    8    8    8    8    7    7    7    7    7    7
    7    7    5    5    5    5    5    5    3    3
      5000      3000      1000      4000      4000
      3000     15000      3000      6000
   .8   .7   .0   .0   .0   .0   .0   .0   .0
   .0   .0   .0   .0   .0   .0   .0   .0   .0
   .0   .0   .0   .0   .0   .0   .8   .0   .0
   .0   .0   .0   .0   .0   .0   .0   .0   .0
   .0   .0   .0   .0   .0   .0   .0   .0   .0
   .0   .0   .0   .0   .0   .0   .0   .0   .0
   .0   .0   .0   .0   .0   .0   .0   .0   .0
   .8   .7   .0   .0   .0   .0   .0   .0   .0
   .0   .0   .0   .0   .0   .0   .0   .0   .0
   .0   .0   .7   .8   .0   .0   .0   .0   .0

  REMAINDER OF COUNTERMEASURE EFFECTIVENESS DATA
            HAS BEEN OMITTED
```

TABLE III-4
SAMPLE INPUT FILE
(GENERALIZED MODEL EXAMPLE)

```
DATA SET FOR PROGRAM CSI
ENTER HERE:
PARAMETERS      REPRESENTING                    FORMAT
T,N             NO. OF OBJECTIVES               2I5
                NO. OF COUNTERMEASURES
M(K)            NO. OF THREATS PER              (10I5)
                OBJECTIVE K
ABSWT(K)        REL. WT. OF OBJECT. K           (10I5)
COST(J)         COST OF COUNTERMEASURE J        (5I10)
MATRIX(K,I,J)   COUNTERMEASURE MATRIX          (10F5.1)
BEFORE ENTERING ANY OF THE PARAMETERS ABOVE, ENTER
ON THE NEXT LINE THE ALGORITHM DESIRED (E.G. G1,
G2,G1BAR,G2BAR).  IF YOU ARE USING A PRE-SELECTED
SEQUENCE, ENTER *PRESET* ON THE NEXT LINE.  IF YOU
WANT ONLY THE COMPUTER SECURITY INDEX
CALCULATED, ENTER *CSI ONLY* ON THE NEXT LINE
PRESET
    4   19
(IF YOU ARE USING A PRE-SELECTED SEQUENCE OF
COUNTERMEASURES, I.E. YOU HAVE ENTERED PRESET
ABOVE, THEN ENTER THAT SEQUENCE ON THE NEXT LINE
IN FORMAT (10I5).  OTHERWISE, CONTINUE ENTERING
PARAMETERS AS DIRECTED ABOVE.)
    4   17    8    5   19    6   11    2    1   15
   14   18   16   12   13    7    3   10    9
    4    8    6    2
    8    7    5    3
      5000      5000      1000      20000      8000
      6000      1000     15000        300       300
      5000      3000      1000       4000      4000
      3000     15000      3000       6000
  .0   .0   .0   .0   .0   .0   .8   .0   .0   .0
  .8   .7   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .8   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .8   .0   .0
  .8   .8   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0
  .8   .0   .0   .0   .0   .0   .0   .0   .0   .0
  .0   .0   .0   .0   .0   .0   .0   .0   .0
```

REMAINDER OF COUNTERMEASURE EFFECTIVENESS DATA
HAS BEEN OMITTED