Copyright © 1977, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

FUZZY COMPUTER SECURITY METRICS:

A PRELIMINARY REPORT

by

L. J. Hoffman and D. Clements

Memorandum No. ERL-M77/6

27 January 1977

UCB-CS-76-42

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

FUZZY COMPUTER SECURITY METRICS:

A PRELIMINARY REPORT

Lance J. Hoffman and Don Clements

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley

ABSTRACT

The definition of a basic security system as a 5-tuple is set forth and then the concept of a covered security system, where at least one security measure exists for each identified penetration path, is briefly discussed. A formal model is presented which focuses upon the interactions of the security measures in a computing system with the threats they combat and with the objects they protect. It is argued that "resistance" (security measure effectiveness) must be combined with some measure of threat probability and object value at each interaction point in the system to arrive at that measure's contribution to overall system security.

The model systematizes the resistance, probability, and value measurement process. The problem of imprecision in these measurements is examined. It is suggested that these security elements are too complex for reliable quantification except in a limited number of cases and the use of human evaluators employing linguistic (as opposed to numeric) measurement tools is proposed. This linguistic measurement process is based upon certain features of fuzzy set theory which substitute words for numbers in the measurement of interactions and quantities which are inherently complex and imprecise.

This research was sponsored by the National Science Foundation Grant MCS76-09214.

I. BACKGROUND

In the last several years, methods for controlling security in computer systems have become widely known. While in 1972 there were only one or two books and three bibliographies on computer security and privacy, there are now at least 14 books and 6 bibliographies on the topic. In addition, knowledge of the techniques and problems involved is much more widespread, as one might guess from the proliferation of literature. Expanding public concern with the problem is evidenced by a great deal of federal, state and local legislation (US 1974, NASIS 1974, BERKELEY 1974, CBEMA 1975). Governmental agencies such as the National Bureau of Standards, the Defense Department, and the National Science Foundation are all sponsoring research efforts in the area, as have some private manufacturers (SAFE 1974, SALTZER 1974).

The upshot of all this work is that a significant part of the computing community is now aware of techniques for maintaining security in computer systems. Unfortunately, the question of how to measure the costs and effectiveness of the various security methods is still largely unexplored. Only recently has there been any reliable work done on costs of privacy transformations or authentication methods, and we have only scratched the surface in investigating metrics for security systems (HOFFMAN 1974). While some more formal work has begun (BELL 1973, ANDREWS 1974, WALTER 1974, HARTSON 1975), there has so far been very little useful spinoff of this theoretical work to practical security decisions. Our ongoing efforts in the development of computer security metrics are discussed in this paper.

I.1. Security Models and Metrics

Introduction

We are currently studying the use of a set-theoretic security model as a vehicle for evaluating data processing installations with respect to security. After first examining the current state of security modeling and the need for some sort of yardstick in evaluating security systems, we present a set-theoretic model which represents our view of the important security relationships in computer systems.

We feel that imprecise measurements are required due to the complexity of security systems and the presence of the human component throughout these systems. We further believe that an application of fuzzy set theory to the imprecision problem will result in a viable, though "fuzzy" rating system. The major features of this theory are outlined in the discussion of the rated security system model below.

Finally, we enumerate some specific research problems we intend to pursue. Our work on these problems is expected to result in the design of a software rating system to be employed by security auditors in the evaluation of system security in data processing installations.

Need for Security Metrics

One of the most difficult decisions for the data processing manager has been security investment. How much of the data processing equipment budget should be allocated to the purchase and maintenance of security features? The decision process remains largely subjective because there currently exists no way of objectively measuring system security. It has sometimes been argued (often by designers of military systems) that security must be a binary condition - a system is either secure or completely insecure. However, it is highly unlikely that many commercial computer users could afford total security even if it were possible to achieve that state. Indeed, total security may not be desirable if efficiency of data processing greatly suffers as a result.

Nevertheless, security measures are necessary in a rapidly increasing number of commercial installations. Sound business practice requires the protection of corporate assets, including both the physical computing equipment and the information therein. Additionally, increasing national concern over privacy and security in computer systems (U.S. 1974) indicates that most users will become even more concerned with security issues in their selection and operation of data processing equipment.

Given the need for security features, this concern remains for the data processing manager: how much more security will I receive if I choose System A over System B? Ideally, this additional amount of security should be quantifiable so that cost-effectiveness tradeoff analysis may be performed. Currently, we are a long way from such precise quantification of security measures. Indeed, we believe that such precision will never be completely attained. There are too many human elements in security systems and "people cannot be 'proven secure' in a non-Orwellian world" (HOFFMAN 1974).

We may, however, attempt to formalize a methodology for use in evaluation of a security system. We can strive to normalize and place bounds upon the imprecision and subjectivity currently employed by computer system evaluators. We propose the development of such a methodology based upon a model described in the following section. We return to the precision issue later.

The Basic System Model

As a first step in the design of security metrics we describe an abstraction of a data processing installation's security system. The description takes the form of a set-theoretic model (the basic security system).

We are currently augmenting this model with certain properties best described using fuzzy set theory.[1]

To date, the application of mathematical modeling to studies of security issues has focused upon software structures functioning within or in conjunction with the operating system in multi-user installations. Such structures function to control user access to information and computing resources (WEISSMAN 1969, LAMPSON 1971, GRAHAM 1972, HSIAO 1974).

Two more recent models are worthy of note. Hartson characterizes a security system as a 5-space (HARTSON 1975). His access control mechanism is concerned with monitoring and granting access requests which result in changes of the system state within the 5-space. His implementation is for use in data-base management applications. Harrison has presented a model of a system with information sharing (HARRISON 1974) and has shown that the owner of information may lose control of access to that information after granting access to a limited number of users. He has proven that the question of guaranteeing that an unreliable user will not pass on an access right to someone unknown to the original owner is undecidable in general.

Our model addresses a broader range of security problems. We are interested in all of the ways in which data of computing resources may be misappropriated. Our view of the computing environment is inspired by (TURN 1974). His conception of the function of a security system is illustrated in Fig. 1. While Turn is interested in the design of security systems subject to user needs and external constraints, the measurement problem can be viewed as the evaluation of the threat domain-security system-protected domain interface. We adopt a narrower view of the system

[1] The properties of fuzzy sets have been studied extensively in the literature (ZADEH 1965, BELLMAN 1970, ZADEH 1973). We will describe fuzzy sets more fully in the next section.

SECURITY ENVIRONMENT

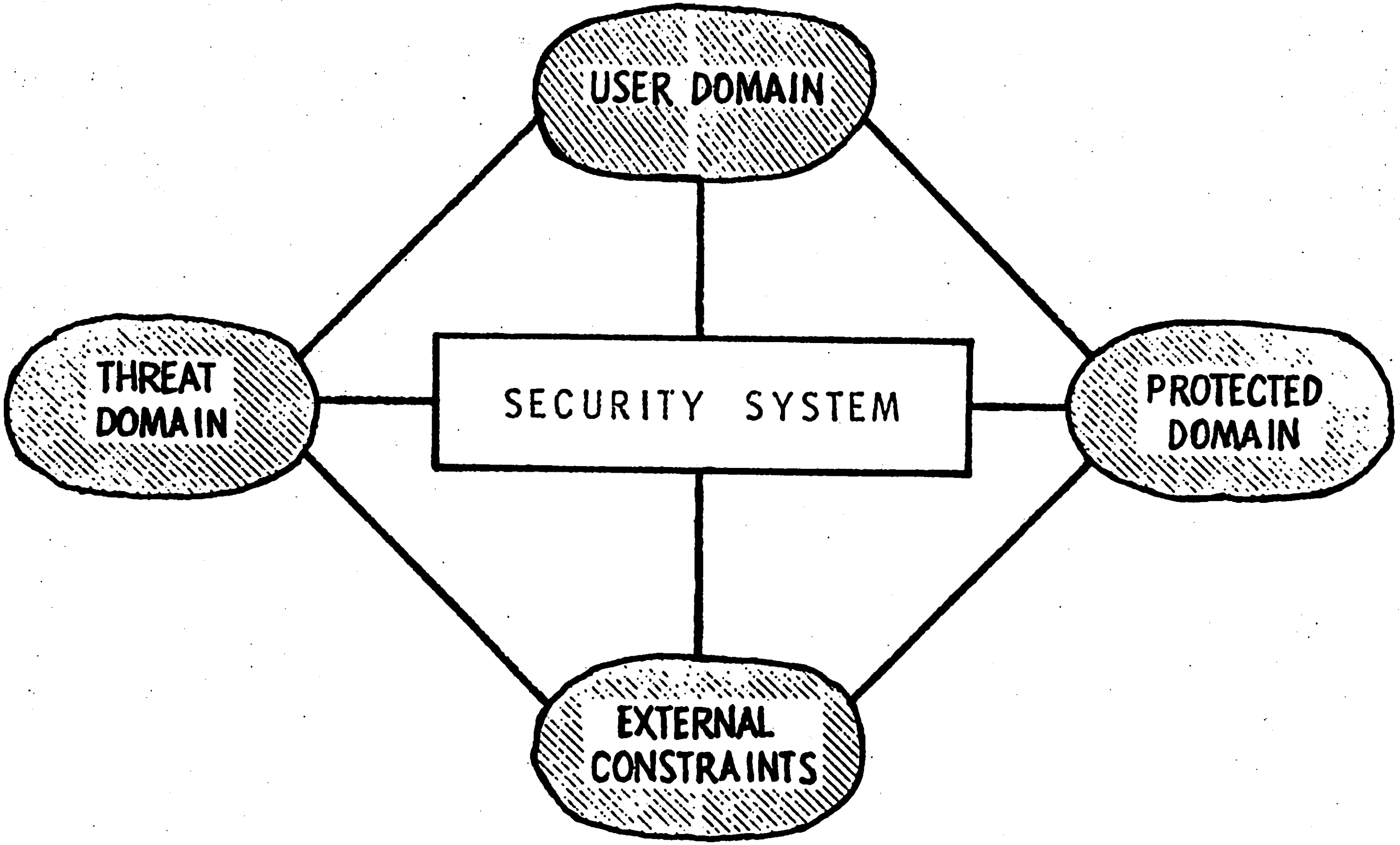

Fig. 1. (from (TURN 1974))

The Rand Corporation
Santa Monica, CA.

environment by eliminating the "user domain" and the "external constraints" from consideration. The objective is to pinpoint each security area, evaluate the protection measure there for effectiveness, and then analyze its contribution to the security of the entire computing system.

Our model focuses upon those resources within computing systems which are vulnerable to some security threat. In any non-trivial data processing system there exists a large variety of such resources, including (but by no means limited to) confidential data, proprietary programs, hardware devices (diskpacks, terminals, etc.), the operating system, processing time, and even elements of the security system itself (e.g., the password file). We group these elements as the set of <u>security objects</u> - O.[2] These objects display a common characteristic: each possess a <u>loss value</u> to its owner which may or may not be quantifiable.

Associated with each security object is a number of activities which a potential intruder may employ to gain unauthorized access to that object. One can attempt to enumerate all of the potential intrusion activities against all of the security objects to form the set of <u>security threats</u> -T. Some common security threats are: wiretapping, exploiting operating system trapdoors, reading core residues, theft, natural disasters, etc. The common characteristic of the threat set is a <u>probability of occurrence</u> associated with each threat. In any real computing system, these probabilities will be calculable with only a limited degree of precision.

The object-threat relations form a bipartite directed graph (Fig. 2).in which edge $\langle t_i o_j \rangle$[3] exists if and only if t_i is a viable means of gaining access to object o_j. It should be noted that the relation of threats to objects

[2] Throughout this discussion, the following notation is observed: an upper case letter (A) names a set and is underscored (<u>A</u>) if the set is fuzzy. Lower case letters (a) denote set elements.

[3] We use $\langle\ \rangle$ to emphasize that the structure is an <u>ordered pair</u>.

is not 1 - 1; a threat may compromise any number of objects and an object may be vulnerable to more than one threat. The goal of the security game is to "cover" each edge of the graph of Fig. 2 by erecting a barrier to access along that path.

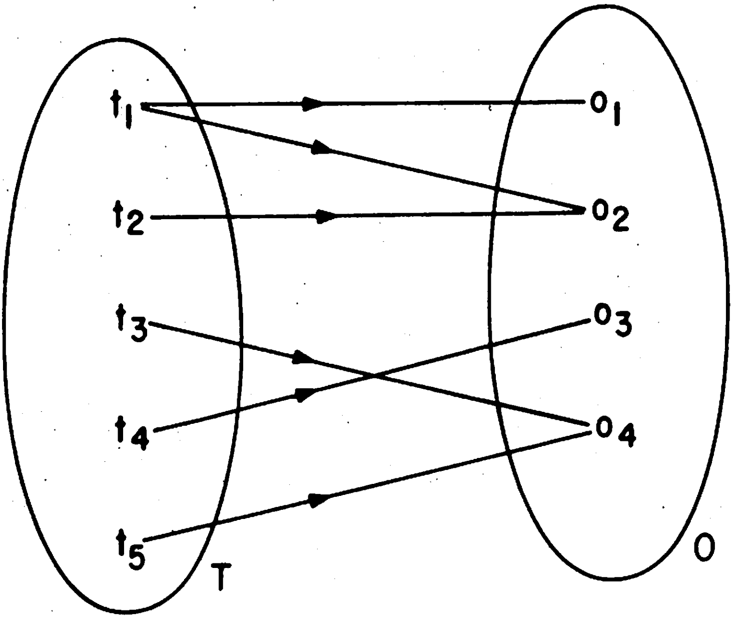

Fig. 2. The threat-object relation

A third set completes our simple model. Its members are the security measures - M employed as protective devices in the computer system.[4] Ideally, each m_k would eliminate some edge $\langle t_i o_j \rangle$ from the graph of Fig. 2. In reality, a security technique performs a firewall function by presenting some degree of resistance to a penetration attempt. This resistance is a common characteristic

[4] We define measures as synonymous with techniques. A security measure is a means of achieving some protection objective. It should not be confused with security metrics, our term for the measurement of the degree to which all security measures enhance system security.

of the members of M. Some examples of security techniques are: crytography, passwords, locks, audit logs, overwriting core residue, etc.

The set of security measures transforms the bipartite digraph of Fig. 2 into the tripartite graph of Fig. 3. In a "protected" system all edges are of the form $\langle t_i m_k \rangle$, and $\langle m_k o_j \rangle$. Any edge of the form $\langle t_i o_j \rangle$ identifies an unprotected object. It should be noted that a single security technique may counter more than one threat and/or protect more than one object (e.g., m_4 in Fig. 3). We also wish to emphasize that the absence of an edge $\langle t_i o_j \rangle$ does not guarantee complete security (although the presence of such an edge guarantees potential compromise unless, of course, the probability of occurrence of t_m is equal to zero).

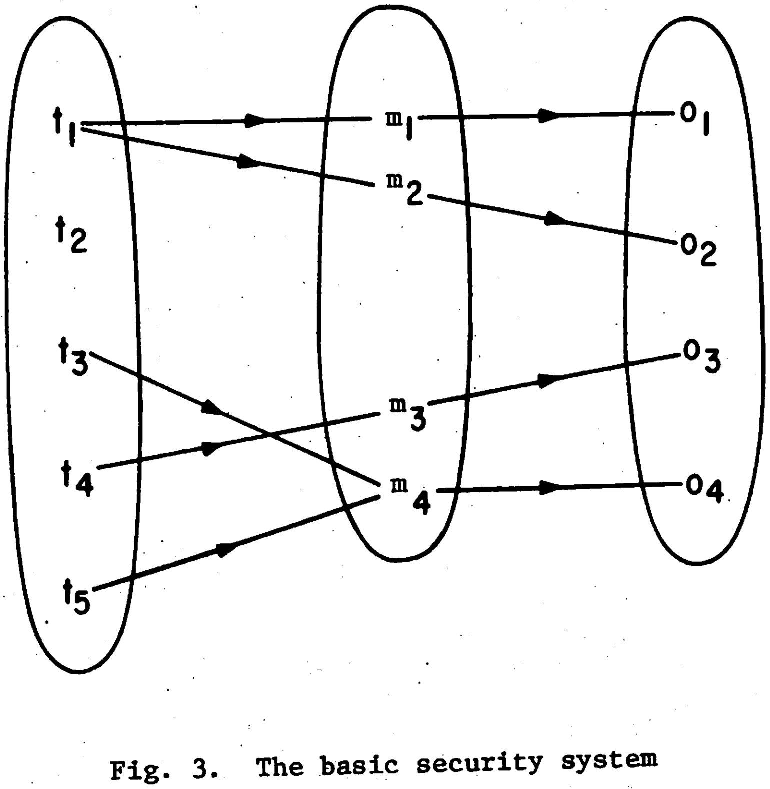

Fig. 3. The basic security system

In summary, a basic security system is a 5-tuple:

$S = \{O,T,M,V,B\}$

where:

O is a set of security Objects

T is a set of security Threats

M is a set of security Measures

V is a set of Vulnerabilities, a mapping of $T \times O$ to a set of ordered pairs $V_\ell = \langle t_i, o_j \rangle$ the penetration paths in the system.

B is a set of Barriers, a mapping of $V \times M$ (or $T \times O \times M$) to a set of ordered tuples: $b_\ell = \langle t_i, o_j, m_k \rangle$, the security points in the system.

We define a covered system as one in which a security measures exists for each identified penetration path. In such a system $\langle t_i, o_j \rangle \in V$ implies $\langle t_i, o_j, m_k \rangle \in B$. If this correspondence does not hold, then o_j is unprotected for some j.

Although our model appears superficially similar to (HARTSON 1975) and other protection models, the underlying elements and relations described greatly differ. Hartson is concerned with access demands and dynamic changes in system state as a result of granting or denying requests. Our model is static but broader in scope. Access control by the data base management system - his object of investigation - would be one barrier in our model. We are also concerned with evaluating the effectiveness of an implementation such as Hartson's and its contribution to the overall security of the computing system.

Hartson's goal is the design of a family of protection specification languages which enable system security administrators to specify ownership, sharing, authorization, access history and other housekeeping parameters in a functioning data base system. Our purpose is to aid the system manager in a post-design security audit. While he focuses upon the user-data base interface, we are concerned with this only in the sense of being interested in its resistance to penetration.

The Problem of Imprecision

If every security measure in our model system possessed an infinite resistance to penetration, our task would be finished. Assuring coverage would be sufficient to assure absolute security. However, real-world security measures present only a limited amount of resistance - passwords have a finite length, a steel door may be cut given sufficient time, etc.

Most security techniques do not lend themselves to quantification.[5] In such instances we must rely on human judgement to provide an approximate measure of this resistance. The problem is further aggravated when we attempt to combine these individual resistance values to obtain an overall system security rating. The individual features often interact;[6] usually these interactions are not well understood. One suggested rating system (HOFFMAN 1974) requires the rater to assign a numerical value to his or her estimate of component resistance. A linear weight and score method is used to producc relative rankings. But, we are now convinced, an evaluation of the security of a given computer system which is based upon numerical tech-

[5] What is the numerical expectation that a user will lose a security badge?

[6] For example: the effectiveness of a password scheme may depend upon the protection mechanism which safeguards the password file.

niques alone cannot be meaningful. The precision of a numerical ranking is inconsistent with the complexity of the data processing installation when viewed as a system. The statement: "XYZ computer installation is .65 secure," while possibly consistent within the framework of some rating system, has little meaning to the outside observer. Further, it is bound to generate more than a little skepticism. Conceptualizing a .65 secure system is much more difficult than conceptualizing, say a .65 full cup of coffee.

Nevertheless, it is possible to make meaningful measurements of the security of a system relative to another or relative to some idealized secure installation. We suggest that the appropriate structure for the expression of such measures is the *linguistic variable* (ZADEH 1973) - a variable which assumes values which are words rather than numbers. A computing system may be represented by a composite linguistic variable (a structure whose components are themselves linguistic variables called attributes). Its attributes might include processing power, cost, amount of storage, security, etc. An exact description of the structure is not necessary here - we are interested only in the security component. This component is a simple linguistic variable which takes on values such as *high*, *low*, and *moderate*. Appropriate modifiers provide finer resolution by allowing values such as *very high*, *somewhat low*, etc. The evaluation of the security of a given data processing system corresponds to the assignment of a value to the attribute *security*. Each value is a fuzzy set[7] whose mem-

[7] A fuzzy set F is characterized by a membership (or compatibility) function μ_F: $[0,1] \rightarrow [0,1]$. If μ_{high} $(0.8) = 0.9$, the 0.9 represents the degree to which a non-fuzzy rating of 0.8 agrees with a fuzzy rating of *high*, Note that the domain of μ is arbitrarily chosen here to be specific to our purpose of defining a rating. The range of a membership function is always [0,1]. See (ZADEH 1965) and (ZADEH 1974).

bers are real numbers in the interval [0,1]. These real numbers correspond to the numerical ratings suggested by (HOFFMAN 1974). We sacrifice the precision of that numeric approach to gain a higher level of confidence that the final "fuzzy" rating is realistic.

The total security rating will be based upon the rater's evaluation of the adequacy of each security mechanism in the system. As mentioned earlier, the effectiveness of a particular security feature may be quite difficult to quantify. Additionally, every evaluation will be somewhat subjective. For these reasons, we will redefine a *security barrier* as a composite linguistic variable. One component of this variable is *resistance to penetration*, a simple linguistic variable which ranges over the same value set as the *system security* variable previously defined. Resistance is taken to mean the degree to which a security technique succeeds in combating the (non-fuzzy) set of threats against which it has been implemented.

Not all security threats are equally likely to occur; nor are the consequences of penetration equally costly, either from threat to threat or from installation to installation. Therefore, each threat associated with a given security barrier will possess an attribute named *probability of occurrence*. This attribute takes on linguistic probabilities[8] as values. Fuzzy probabilities are very appropriate for this type of threat analysis.

The final parameter of interest is the *loss value* of the resource or data under attack. A *rational security policy*[9] will balance probability,

[8] Linguistic probabilities assume values such as *likely*, *improbable*, etc. An introduction to the theory of linguistic probabilities may be found in (ZADEH 1973).

[9] As opposed to a dogmatic policy which requires complete protection of all resources. See (TURN 1974).

loss value, and resistance. These three parameters, when added to our basic system, form a "rated" system model which is the basis of our evaluation procedure.

The Rated Security System Model

Our basic model may now be augmented to include the evaluation parameters introduced above. Each element of the barrier set (B) is a composite linguistic variable $\mathcal{B}_\ell$ with three components. Each component consists of a name and a linguistic value. For example, assume $b_\ell = \langle t_i, o_j, m_k \rangle$ in the basic system. The threat component of $\mathcal{B}_\ell$ in the rated security system has name t_i and probability $\mathcal{P}_\ell$ (where $\mathcal{P}_\ell$ is a linguistic variable).[10] The complete structure of $\mathcal{B}_\ell$ is illustrated in Fig. 4

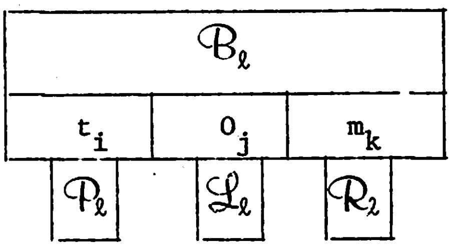

Fig. 4. The security barrier as a composite linguistic variable

Note that the subscripts for threat Probability ($\mathcal{P}$), object loss value ($\mathcal{L}$) and security measure resistance ($\mathcal{R}$) match the subscript of the barrier ($\mathcal{B}$) rather than the individual threat object, and security measure components of the barrier. This is to emphasize that these components are evaluated in the context of the specific barrier which they form.

As a qualitative example of this interdependence, consider a file con-

[10] We will use script capital letters to represent linguistic variables which are neither fuzzy or non-fuzzy sets but rather variables which may be assigned values which are fuzzy sets. See (ZADEH 1973) for a more formal description of fuzzy and non-fuzzy variables.

taining a proprietary program belonging to a software engineering firm. The loss value of the program may be quite high in the context of a threat of theft by a competitor who may exploit its commercial value. On the other hand, the file may have only a moderate-to-low loss value in the context of a threat of accidental erasure (especially if a backup copy exists).

The compatibility function (μ) relates the linguistic value (high, low, very high, etc.) assigned to the lingustic variables ($\mathcal{P}, \mathcal{L}, \mathcal{R}$) to its corresponding base variable.[11] It may be discrete (a tabulation) or continuous. An example of a continuous compatibility function for the value high in reference to probability ($\mathcal{P}$) is given in Fig. 5a. We emphasize that μ_{high} is not a probabilistic function. Rather, it relates the various numerical probability values (0.0, 0.2, 0.9, 1.0, etc.) to our conception of a high probability.

As mentioned in our description of the basic security system, there may be unprotected objects ($\langle t_i, o_j \rangle \in V$ and $\langle t_i, o_j, m_k \rangle \notin B$ for all k) in a given system. For simplicity and consistency within the model we add a null security measure m_o[12] to the set M. The B mapping is augmented so that each unprotected object will be given the null measure. Then for each $\langle t_i, o_j \rangle \in V$ where $\langle t_i, o_j, m_k \rangle \notin B$ for all k, the barrier $\langle t_i, o_j, m_o \rangle$ is added to B.

The Evaluation Process

The system rater assigns linguistic values (high, low, very high, moder-

[11] A base variable is a numeric (non-fuzzy) variable which ranges over an interval appropriate to the linguistic variable in question. Thus, in the case of $\mathcal{P}$ (i.e., probability) the basic variable range would be [0,1] while for $\mathcal{L}$ (loss value the range might be [\$0, \$1,000].

[12] Any $\mathcal{B}_\ell = \langle t_i, oj, m_\theta \rangle$ will have (by definition) $\mathcal{R}_\ell$ = none where none is totally compatible with a numeric resistance of 0 (zero). In the notation of (ZADEH 1973), μ_{none} (0) = 1, μ_{none} (i) = 0 for all i in the range of the associated base variable.

ate, etc.) to the component variables $(\mathcal{P}_\ell, \mathcal{L}_\ell, \mathcal{R}_\ell)$ at each barrier in the system. These values determine the *contribution* of the barrier to total system security. Informally, the combination of probability and loss value yields the importance (weight) of the barrier in the composite rating. The resistance value determines the degree to which overall system security is enhanced or degraded.

Naturally, we expect differing opinions among the raters making these evaluations. We intend to incorporate normalization and consistency checking procedures as outlined in the following section.

Further Research

Our current work is progressing along the following lines.

1) Fuzzy Rated Security System

We are refining and adding detail to our description of the fuzzy model (the rated security system) The central problem here is the design of the composition formula for combining the values at each barrier to determine the total system security rating. Should the fuzzy analog of the linear weight and score method be used? What is the nature of the importance (weighting) function? For example, a low probability threat against an object with a high loss value is intuitively of greater concern that a high probability threat against a low value object. It seems reasonable that the weighting should differ in each case, bnt the amount of difference is not obvious.

2) Linguistic Value Selection

We are formulating a canonical set of terms to be used as linguistic values in the rating process. The primary terms *high*, *low*, and *moderate*, in conjunction with linguistic modifiers such as *very*, *extremely*, etc. are a starting point. "Standard" compatibility functions will be developed for the primary terms. We are formulating the process by which modifiers may act upon the compatibility functions of primary terms to creat new fuzzy sets. For example, very applied to μ_{high} (see Fig. 5a) would yield $\mu_{very\ high}$ as shown in Fig. 5b.

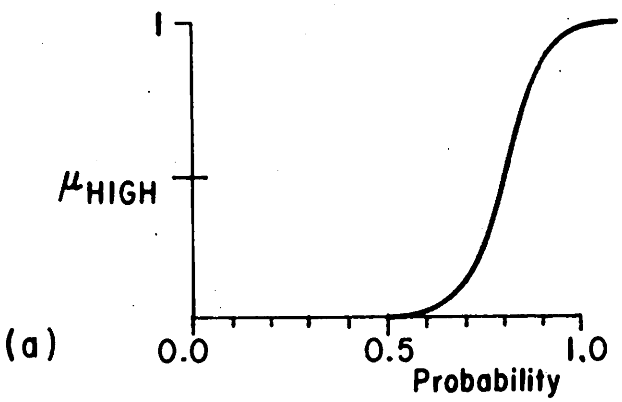

Fig. 5a. Compatibility function of High Probability.

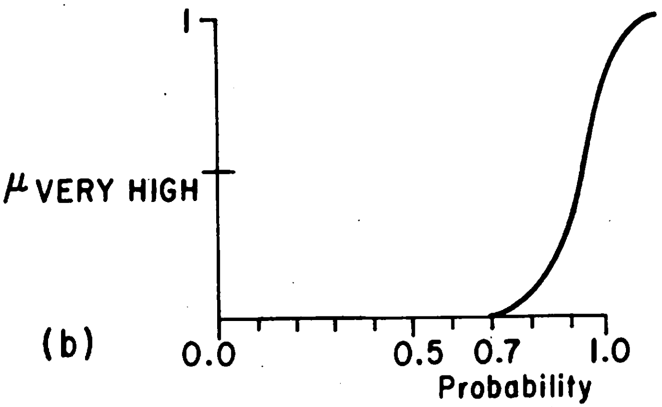

Fig. 5b. Compatibility function of Very High Probability.

3) Linguistic Bias

The "standard" functions mentioned above may not agree completely with each rater's preconception of the canonical linguistic terms. We intend to explore methods of introducing linguistic bias to align each function with the current rater's definition of the various values.

4) Software to Calculate System Ratings

We plan to design a software system for calculating security ratings by simulating the linguistic value fuzzy sets using non-fuzzy hardware. The program will tabulate the importance (weight) and resistance fuzzy values at each barrier, calculate the composite compatibility function and perform the linguistic approximation[13] to arrive at the linguistic value which represents the system rating.

As a qualitative and crude example, consider a system composed of two barriers. $\mathcal{B}_1 = \langle t_1, o_1, m_1 \rangle$ where t_1 is a software bug in user 1's program which overwrites his or her private data file, o_1 is the data file, m_1 is optional read only protection for the file system. $\mathcal{B}_2 = \langle t_2, o_1, m_2 \rangle$ where t_2 is user 2 browsing through the files of user 1, o_1 is as previsouly defined, m_2 is password protection of private files. Assume that rater 1 assigns the following values at the barriers:

$\mathcal{B}_1$: $\mathcal{P}_1$ = fairly high
$\mathcal{L}_1$ = high
$\mathcal{R}_1$ = very, very high

$\mathcal{B}_2$: $\mathcal{P}_2$ = moderate
$\mathcal{L}_2$ = very high
$\mathcal{R}_2$ = moderate

Qualitatively, $\mathcal{B}_1$ by itself would suggest a near perfect rating. How-

[13] Linguistic approximation is the process of determining the linguistic value which most closely agrees with a given composite corpatibility function. See (ZADEH 1973).

ever, the damage could be great at $\mathcal{B}_2$ even though the security there is probably appropriate for the threat involved. We would expect a total security rating of high or perhaps merely fairly high for this trivial file system.

5) Ratings of Real-world Systems

Finally, we intend to compile sample "ratings" of some "real-world" data processing systems to exercise the rating program and test the validity of this approach to security measurement.

Summary

While we are firmly convinced of the need for a set of security metrics, we are just as convinced that, given the current state of the art, ratings must be based upon human judgement. There are just too many complex and poorly understood aspects of the security problem. We are skeptical of numeric rankings based upon qualitative (and often subjective) evaluation by human auditors. One definition of the dilemna is the principal of incompatibility: as system complexity increases, analytical precision decreases (paraphrasing (ZADEH 1973)).

We therefore believe the present limits of security engineering dictate an evaluation methodology which is exhaustive in locating system vulnerabilities, but somewhat imprecise in estimating the relative "goodness" of various security features and their contribution to the overall security design goal.

There is some existing work aimed at formalizing security design guidelines (TURN 1974) and much literature concerning abstraction of protection mechanisms within operating systems and data base management systems (HARTSON 1975, HARRISON 1974, HSAIO 1974, LAMPSON 1971, SALTZER 1975). We know of no other attempts to apply the methodology of mathematical modeling and fuzzy set theory to the measurement of computer security. With the exception of (HOFFMAN 1974), possible mechanisms for comparative rankings of commercial security systems have not been explored. We believe fuzzy security ratings will provide an important first step in the development of security metrics.

III. REFERENCES

(ANDREWS 1974) Andrews, G. R., "COPS - A Mechanism for Computer Protection," Technical Report 74-06-23, Computer Science Department, University of Washington, Seattle, June 1974.

(BELL 1973) Bell, D. and LaPadula, L. J., "Secure Computer Systems: A Mathematical Model," MTR-2547, the MITRE Corporation, Bedford, Mass., November 1973.

(BELLMAN 1970) Bellman, R. E. and Faden, L. A., "Decision-Making in a Fuzzy Environment," Management Science, Vol. 17, pp. B-141-B-164, 1970.

(BERKELEY 1974) Ordinance Number 4732-N.S., Social Impact Statement for Automated Record Keeping Systems, Berkeley, California City Council, 1974.

(CBEMA 1975) Computer and Business Equipment Manufacturers Association, Periodical Lists of State Legislation on Privacy and Security.

(GRAHAM 1972) Graham, G. S. and Denning, P. J., "Protection Principles and Practice," Proc. 1972 SJCC, pp. 417-429, May 1972.

(HARRISON 1974) Harrison, M. A., Ruzzo, W. L., and Ullman, J. D., "On Protection in Operating Systems," submitted to ACM SIGOPS, Operating Systemw Review, 9, 5 14-24, 1970.

(HARTSON 1975) Hartson, J. R. and Hsiao, D. K., "Languages for Specifying Protection Requirements in Data Base Systems (Part 1)," Ohio State University, Computer and Information Science Research Center, Report OSU-CISRC-TR-74-10, January 1975.

(HOFFMAN 1974) Hoffman, L. J., "Constructing Security Ratings for Computer Systems," Proc. IEEE National Telecommunications Conf., December 1974.

(HSIAO 1974) Hsiao, D. K., Kerr, D. S. and Nee, C. J., "Context Protection and Consistent Control in Data Base Systems (Part I)," Ohio State University, Computer and Information Science Research Center Report OSU-CISRC-TR-73-9, Columbus, Ohio, 1974.

(LAMPSON 1971) Lampson, B. W., "Protection," Proc. Fifth Annual Princeton Conference, pp. 437-443, March 1971.

(NASIS 1974) National Association for State Information Systems, Suggested Guidelines for a State Information Practices Act.

(SAFE 1974) "What Every Executive Should Know About Privacy in Information Systems," Project SAFE, State of Illinois, 1974.

(SALTZER 1974) Saltzer, J. H., "Ongoing Research and Development on Information Projection," ACM Operating Systems Review, 8, 3, July 1974.

(SALTZER 1975) Saltzer, J. H. and Schroeder, M. D., "The Protection of Information in Computer Systems," Proc. IEEE 63, 9, pp. 1278-1308, September 1975.

(TURN 1974) Turn, R., "Toward Data Security Engineering," Memo P-5142, Rand Corporation, Santa Monica, California, January 1974.

(U.S. 1974) U.S. Congress, Public Law 93-579.

(WALTER 1974) Walter, K. G., Ogden, W. F., Rounds, W. C., et al., "Primitive Models for Computer Security," EDS-TR-74-117, Case Western Research University, Cleveland, Ohio, January 23, 1974.

(WEISSMAN 1969) Weissman, C., "Security Controls in the Adept-50 Time Sharing System," Proc. 1969 FJCC, 119ff.

(ZADEH 1965) Zadeh, L. A., "Fuzzy Sets," Information and Control, Vol. 8, pp. 338-353, 1965.

(ZADEH 1973) Zadeh, L. A., "The Concept of a Linguistic Variable and its Application to Approximate Reasoning," Memo ERL-M411 Electronics Research Laboratory, University of California, Berkeley, October 15, 1973.

(ZADEH 1974) Zadeh, L. A., "A Fuzzy-Algorithmic Approach to the Definition of Complex or Imprecise Concepts," Memo ERL-M474, Electronics Research Laboratory, University of California, Berkeley, October 11, 1974.