SECURITY IS FUZZY!

Applying the Fuzzy Logic Paradigm to the Multipolicy Paradigm

Hilary H. Hosmer
Data Security Inc.

Based upon computer security work sponsored by ACM SIGSAC,
the U.S. Air Force Materiel Command ESC/ENS, and
the Air Force Cryptologic Support Center AFCSC/SRER.

ABSTRACT

Fuzzy logic is a relatively new paradigm which may radically impact computer security. It can be used in formal methods, in trusted system analysis and design, in measuring the security of systems, and in representing the imprecise human world of policies and inference. The implications are challenging and complicated. This paper reviews basic fuzzy logic concepts, illustrates their use with examples from computer security, and incorporates fuzzy logic into the Multipolicy Machine architecture. It is easier to use tools designed to deal with fuzziness than search in vain for the illusive perfectly secure system.

The Multipolicy Paradigm[1]

Policy-making is a human enterprise, integrating many complementary, contradictory, fuzzy, and changing human values. Every person, as well as every computer system, participates in multiple policy domains. Automated policy systems should model human systems, allow different authorities to change the policies under their jurisdiction and allow *ad hoc* resolution of policy conflicts when two independent policies clash.

Multiple policies occur when there are multiple security goals (such as privacy, confidentiality and integrity), diverse constituents with individual goals and plans (e.g., the members of the European Community), separately evaluated pieces (e.g. OS and DBMS), or a need to adapt to changing situations. *A priori* integration of these policies is often impossible,[2] necessitating a new approach.

H. H. Hosmer 1990, 1991

The Fuzzy Logic Paradigm[3]

"Unquestionably, computers have proved to be highly effective in dealing with mechanistic systems, that is inanimate systems whose behavior is governed by the laws of physics, chemistry and electromagnetism. Unfortunately, the same cannot be said about humanistic systems, [i.e.] systems whose behavior is strongly influenced by human judgement, perception or emotions."[4]

"In order to make significant assertions about the behavior of humanistic systems, it may be necessary to abandon the high standards of rigor and precision that we have become accustomed to expect...and become more tolerant of approaches which are approximate in nature"[5]..."The crux of the problem, really, is the excessively wide gap between the precision of classical logic and the imprecision of the real world."[6]

L. A. Zadeh 1975, 1984

Fuzzy logic offers the rigor of formal methods without requiring undue precision. It also offers alternative methods to handle policy preferences and conflicts. Fuzzy logic "has matured into a wide-ranging collection of concepts, models, and techniques for dealing with complex phenomena which do not lend themselves to analysis by classical methods based on probability theory and bivalent logic."[7]

H. J. Zimmerman 1987

INTRODUCTION

Fuzzy logic is a well-developed set of concepts, techniques, and theorems designed to handle vagueness and imprecision. It is effective on continuous data, such as temperature, on human reasoning, which is usually imprecise, and on very complex systems, where the algorithms are not explicitly known. This paper proposes that fuzzy logic in the broad sense can be useful in computer security to address the problems inherent in complex, multipolicy, human-interacting systems.

Fuzzy logic has been well-worked out by its founder, Lofti Zadeh, and many others, including the Japanese. Zadeh formally developed the issues and established the theoretical framework in a prodigious amount of work from the nineteen-sixties to the present. Eighteen of his key papers appear in a 1987 compilation.[8] Other researchers have published about 15,000 works[9] in the fuzzy logic area exploring both the quantitative and qualitative aspects of vagueness.

Although Dr. Bhavani Thuraisingham of MITRE recommended the use of fuzzy logic for solving problems in database inference,[10] and I recommended its use for modelling non-traditional policies and interfacing more easily with users,[11] fuzzy logic has not yet made substantial inroads into computer security. The name may suggest muddled thinking, and there is little awareness of the many possible applications.

There may be a more fundamental reason for the information security (INFOSEC) community's failure to embrace fuzzy logic. As Ruth Nelson notes, in the current computer security paradigm, the ultimate goal is provable absolute security. This requires logical and mathematical precision. Unfortunately, precision and complexity are inversely related. As long as precision is required, trusted systems cannot be very complex. In the fuzzy logic paradigm, truth may be fuzzy, as it usually is in the real world. This will allow systems of much greater complexity. However, a major shift in thinking is required to concede that much of security is (and always will be) fuzzy and that computer security can benefit from logic designed to handle vagueness and imprecision.

The paper starts with a fuzzy logic primer, applying fuzzy logic to computer security, then applies fuzzy logic to the Multipolicy Paradigm.

FUZZY LOGIC PRIMER

Fuzzy logic extends traditional logic, enabling it to handle approximations and linguistic (i.e., non-numeric) variables. It is a superset of current logic, so all current forms of logic, such as predicate calculus, can be incorporated. Usually, only the objects of manipulation are fuzzy, not the logic that deals with them. The example below adapted from [12] illustrates that doing deductive logic with approximations can be clear, even elegant.

Definitions:
~ means "approximately"
CVB stands for CoVert channel Bandwidth

Premises:

CVB-1 is small
CVB-1 and CVB-2 are ~ equal

Approximate conclusion:

CVB-2 is more or less small

Figure 1. Fuzzy Deductive Logic

"A **fuzzy set,**" wrote Zadeh, "is a class with unsharp boundaries, that is a class in which the transition from membership to non-membership is gradual rather than abrupt."[13]

Foods Americans Eat *{Cheeseburger, Apple Pie, Sushi, Junk food...}* is an example of a fuzzy set. Although some Americans eat sushi, it would never appear on a list of great American dishes and should be counted as only a partial member of the set. Similarly, the set of available intrusion detection devices is fuzzy because devices with related functions, such as monitoring or identification would be included.

Members participate in a fuzzy set to some degree.[14] Conventionally, a number between 0 and 1 is used to show this degree, with 1 being high. The numbers representing the degree of membership are sometimes selected subjectively. For example, foods might participate in the *Foods Americans Eat* set to the following degrees:

Item Name	Degree of Participation
Cheeseburger	.9
Junk food	.4
Sushi	.1
Metal bar	0

Figure 2. Participation in Fuzzy Sets

A **crisp set** is a set in which members clearly belong or do not belong to the set. Crisp sets are just fuzzy sets where the range of possible membership degrees is limited to zero and one.

In computer security, the trusted computing base (TCB) is a crisp set. Code either is or is not in the TCB. The TCSEC mandates a clearly defined boundary between trusted and untrusted code.

If the TCB were a fuzzy set, it would be possible to distinguish between critical and less critical components of the TCB by classifying their degree of membership in the TCB. Figure 3 illustrates that kernel TCB processes would have high degrees of participation in the TCB set, while trusted user applications with limited security functionality would have medium to low degrees of participation. Untrusted code would have zero degrees of participation. The effects of this change in perspective are unclear, but it might simplify both implementation and evaluation of multilayered TCBs.

Code Type	Degree of Participation in Trusted Computing Base
Untrusted Applications	0
Trusted Applications	.50
Trusted System Software	
I&A	.90
Trusted path/window	.88
Audit	.85
Other	.80
Reference monitor	.999

Figure 3. Fuzzy Set Participation

In computer security privilege currently is a crisp set. A subject either has a privilege or does not. This crispness makes it difficult to model any in-between conditions, such as having the privilege under certain circumstances, or being in the process of losing a privilege. Fuzzy privileges might be useful.

Anyone who has worked on an INFOSEC standards committee realizes that, upon examination, most of the key INFOSEC concepts are fuzzy. When the actual members of any set, such as subjects or objects, are studied, their diversity and complexity become apparent.

INFOSEC candidates for fuzzy sets include:

Subjects
Objects
Secrets
Risk
Grave risk
Trusted computing base
Security violations
Encryption techniques
Policy objectives
Formal security policy models
Informal policy models
Evaluation procedures
Networks
Systems

Prototypes are a useful notion for simplifying the complexity of the real world. Each of the INFOSEC concepts listed above is, in fact, a fuzzy logic prototype. Eleanor Rosch[15] developed the prototype concept in her research exploring intuitive notions of the fuzziness of classes. She did a number of experiments asking students to rate how much something participated in its class. For example, she asked students to rate a number of birds, including robins, ostriches, and penguins, as to their degree of "birdness". The students easily gave each bird a rating which reflected its perceived degree of participation in the bird class. Because the robin was consistently perceived to be more like a bird than the others, Rosch called it a "prototype". Penguins, perhaps because they don't fly, were perceived to be less birdlike than the others. Rosch illustrated that classes are intuitively fuzzy once one looks at the real members of the class, but that via prototypes people still maintain a simple and useful concept of the characteristics of a class.

From the fuzzy logic perspective, our INFOSEC concepts of subject, object, domain, TCB, assurance, etc. are prototypes enabling us to maintain a simple and

useful concept in spite of the diversity and complexity of the members of these fuzzy sets. Classes of policies, like "discretionary" and "non-discretionary", or "integrity", "assured service" and "confidentiality" are also prototypes. We believe we understand what we are talking about until confronted with a variety of real examples. Then the underlying fuzziness becomes apparent.

To illustrate, contrast the current concept of subjects as represented in the TCSEC (Figure 4a) with the fuzzy concept of subjects (Figure 4b).

Topic: Subjects

TCSEC definition: Process/domain pair

Common definition: Active system entity, i.e., persons, processes, and devices which change the system state

Candidate Subjects: System Security Officer, console operator, workstation user, corporate president, U.S Congressional Representative.

Normally, the corporate president and the member of Congress would be classified as "Not subjects" since they rarely physically touch a system. However, they often play a significant role in approving security policies which are implemented in the system. Policy-makers impact the system state whenever they change the policy. Should they be classified as subjects or not-subjects?

Figure 4a. TCSEC Subject

Fuzzy logic allows infinite gradations of subjects. This might be useful in access control as well as system tailoring. See Figure 4b.

Topic: Fuzzy Subjects

Definition: The set of all entities which act to some degree to change the system state.

Role	**Degree of Participation**
SSO	.99
console operator	.90
workstation user	.8
corporate president	.4
Congress representative	.01

With fuzzy logic there can be separate rankings for subjects which affect the system state and those who effect the meta-system state, i.e., policy. The enterprise can select which policy maker (e.g., corporate president, Congress representative) has more impact in which domain and which privilege-holders (e.g., SSO, console operator) impact the system more than others.

Figure 4b. Fuzzy Subjects

Modelling Reality

The abstractions of formal logic often seem "unreal" because they don't capture the complexity and continuous nature of the "real" world. Fuzzy logic provides a way to formally model imprecision and vagueness, making it possible to incorporate much more of the "real" world into our security models.

Wherever a **continuum** is found, fuzzy classes are likely to be appropriate. "Risk" and "grave risk to national security" are fuzzy points on a fuzzy risk continuum underlying our DOD classifications.[16] Degrees of secrecy and of integrity are, too. Other INFOSEC continua include:

Availability
Assurance
Covert channel bandwidths
Efficiency
Emanations
Violation levels
Information flow variable security levels.

Items on a continuum are measured, not counted. Hence, NSA has scales for measuring the strength or speed of encryption algorithms[17], and the NCSC's Trusted Computer System Evaluation Criteria (TCSEC) provides a scale for measuring the security of systems and the width of covert channels.

In traditional logic, **breakpoints** divide any continuum into discrete points. For example, everything above 78F degrees might be "hot" while everything below 78F degrees is "not hot". The NCSC provides such breakpoints for covert channels, defining what

bandwidths are tolerable, which must be monitored, and which must be closed. The familiar TCSEC ratings (A1, B3, B2, B1, C2, C1) describe discrete breakpoints.

Breakpoints exaggerate differences, distorting the continuous nature of underlying phenomena. For that reason, the ITSEC and Federal Criteria (FC) efforts propose to open up the TCSEC clusters so that trusted systems can use many more points on the assurance and functionality continua. Security profiles permit users to define their own combinations with great flexibility. Similarly, fuzzy logic preserves the underlying continuum, and allows application of names to fuzzy points and regions of the continuum. Figure 7 illustrates a fuzzy approach to the problem of defining acceptable covert channel bandwidths.

Covert Channel Bandwidths (CVB)

CVB	Participation in LARGE CVB set
10 bits per second	.001
100,000 bits per second	.01
100,000,000 bits per second	.9

CVB	Participation in SMALL CVB set
10 bits per second	.99
100,000 bits per second	.05
100,000,000 bits per second	.0001

Figure 5. Covert Channel Bandwidths are Fuzzy

A continuum is not required for fuzzy classes to be useful. Fuzzy classes also (and very frequently) appear when the underlying set is finite or discrete.[18]

The Multipolicy Paradigm permits ad hoc **conflict resolution** of conflicting policies. In both fuzzy[19] and traditional logic, conflict resolution involves goals, constraints, decisions, and consequences. These are rarely known precisely in the real world, . "I would like to earn more than about $60,000 per year" is a typical goal. It is also a **fuzzy goal**, defined by Zimmerman as an objective which can be characterized as a fuzzy set in an appropriate space[20]. Some examples of fuzzy goals follow.

Fuzzy Goal:	Salary ~>= 60000
Fuzzy Goal:	More challenging work

Figure 6. Fuzzy Goals

A **fuzzy constraint** is also an objective which can be characterized as a fuzzy set in an appropriate space.[21] For example, "I want to live within 20 miles of my job". "The house should cost in the vicinity of 200,000 dollars".

Fuzzy Constraint:	Commute ~<= 20 miles
Fuzzy Constraint:	Cost ~= $200,000

Figure 7. Fuzzy Constraints

A decision is basically a choice or a set of choices drawn from the available alternatives. A **fuzzy decision** is the fuzzy set of alternatives resulting from the confluence of the goals and constraints.[22] For example, the set of jobs which offer more challenging work, a salary over approximately $60,000, a short commute, and homes in the $200,000 more or less range. This could be the intersection of the fuzzy goal and constraint sets, and a maximizing function could be used. (Other functions might be used for other kinds of problems).

If some of the goals or constraints are more important than others, the decision might be expressed as a convex combination of the goals and the constraints, with the weighting coefficients reflecting the relative importance of the constituent terms.

Some fuzzy INFOSEC goals might include:

Multilevel security
High assurance
Inference control

Fuzzy INFOSEC constraints might include:

Verified software
Access control rules
Laws and regulations

Some fuzzy logic papers are relevant to work on conflicting security policies. Janus Kacprzyk and

Andrzej Straszak's paper, "Application of Fuzzy Decision-Making Models for Determining Optimal Policies in 'Stable' Integrated Regional Development",[23] provides examples of policy statement representation with fuzzy goals, fuzzy constraints, and fuzzy decision-making models. Zimmerman's work[24] in decision-making using fuzzy logic is very useful, especially the many approaches to multi-criteria decision-making in ill-structured situations. Other good articles on decision-making with fuzzy logic, include Chang's "On Risk and Decision-Making in a Fuzzy Environment".[25] and Asai, Tanaka, and Okuda's "Decision-Making and Its Goal in a Fuzzy Environment".[26] Constantin Negoita explored expert fuzzy systems, appropriate for policy rules in [27]. Several authors, including Kandel[28] and Kacpryzk, apply fuzzy set theory to policy analysis and information systems.

To manipulate fuzzy sets, **fuzzy operators** and rules are needed. Fuzzy logic extends traditional bimodal logic in many dimensions. For example:

> Traditional logical operators AND, inclusive OR, exclusive OR and NOT produce results which are either TRUE or FALSE. However, fuzzy logic allows a **continuum of logical results**, such as *completely true, true, very true, almost true, more or less true, untrue, and false*.[29]
>
> Fuzzy logic also permits a **continuum of logical operators**. For example, there is an infinite number of possible operators between AND and OR.
>
> Fuzzy logic extends the meaning of such logical operations as negation, disjunction, conjunction, and implication to handle linguistic values, like *almost true*.

Fuzzy logic permits standard arithmetic operations and many other kinds of mathematics on fuzzy objects, with extensions to handle the imprecision. For example:

> The basic operators have been fuzzified. For example, ~<= represents the fuzzified version of <=, and is interpreted as "essentially smaller than or equal".[30]

Linguistic variables[31] are extremely useful in bridging the gap between human discourse and the formal representation or model. While variables traditionally take on numeric or logical values, linguistic variables take on vague verbal values, such as *high, low, fast,* or *slow*.

For example, in Figure 5 the elements of a patient's data record are classified using linguistic variables. Privacy classifications become intuitive when "human" language is used.

Field Name	*Fuzzy Privacy Classification*
Patient Name	low
Patient Address	medium
Insurance Company	low
Diagnosis	high
Medical Record	high

Figure 8. Linguistic Variables

Linguistic variables, for processing, are mapped to numbers between 0 and 1. "Medium" might be the range from .5 to .6, for example.

Hedges, also called **modifiers,** are terms that modify other fuzzy sets. They include terms like *more or less, very, somewhat, several, almost possible,* which alter a set's range. *Very, for example,* concentrates and narrows a set down or, in a graph, shifts it toward the end. Hedges can break down a continuum into fuzzy chunks, like *very cold, moderately cold, slightly cold.* Hedges match human reasoning and terminology. For example, fuzzy logic has been successful in many control applications because it allows intuitive and gradual expressions of the rules for change.

Cruise Control

IF the car is traveling very fast,
THEN slow it greatly.

IF the car is traveling a little fast,
THEN slow it slightly.

IF the car is traveling a little slowly,
THEN increase gas slightly.

Figure 9. Hedges

Similar hedges could be used for intrusion detection, database inference, and monitoring covert channel

activity. In another example, taken from our previous work,[32] hedges reflect the imprecision of ordinary life.

Hospital Policy

Celebrity patient data is to be accessed only by medical personnel with *high* need-to-know.

Normally, patients cannot see their own medical records.

During the day, patient data is mirrored in real-time to a second off-site database.

Figure 10. INFOSEC Hedges

Examples of infosec hedges include *very high* fault tolerance, *very strong* password mechanisms, and *more complex* distributed system security. Linguistic hedges make it easier to express human policies in computer terms. This idea is developed more fully in our paper "Using Fuzzy Logic to Represent Security Policies in the Multipolicy Paradigm", cited earlier.

We turn now to look at the possible uses of these fuzzy logic techniques in the Multipolicy Paradigm.

APPLYING FUZZY LOGIC TO THE MULTIPOLICY PARADIGM

Fuzzy logic techniques which can be applied to the Multipolicy Paradigm include: fuzzy constraints, fuzzy decision-making, realistic policy modelling, hedges, linguistic variables, degrees of assurance, fuzzy covert channel bandwidths, and graded TCB modules.

Constraints are often used in computer security for assuring the integrity of data. For example, there are data type constraints, data value constraints, data range constraints, referential integrity constraints. Fuzzy constraints permit a more real-world model of security-related phenomena. See the story in Figure 11 which cries out for fuzzy constraints.

Fuzzy rules expand the range of rule-based systems, such as MITRE's General Framework for Access Control. Fuzziness permits imprecise principles to be embodied as easily as well-defined ones. Hedges and linguistic variables can also be used to express human policies and to smooth the interface with users.

A True Story

A travel agent was unable to get a rental car for a flight arriving in Columbus Ohio at 6:58 a.m. She booked the client on another flight arriving at 7:05 a.m., and had no trouble getting a car. She then cancelled the second plane reservation and kept the car reservation.

The rental car agencies in Columbus all opened at 7:00 a.m. and the reservations computer was programmed to refuse cars for flights arriving before 7 a.m. The computer didn't care that arrival times are usually fuzzy and passengers need some time to get from the gates to the office. Fuzzy rather than crisp constraints were needed in the reservations system.

Jim Smith via Sergei Ovchinnikov, 1993

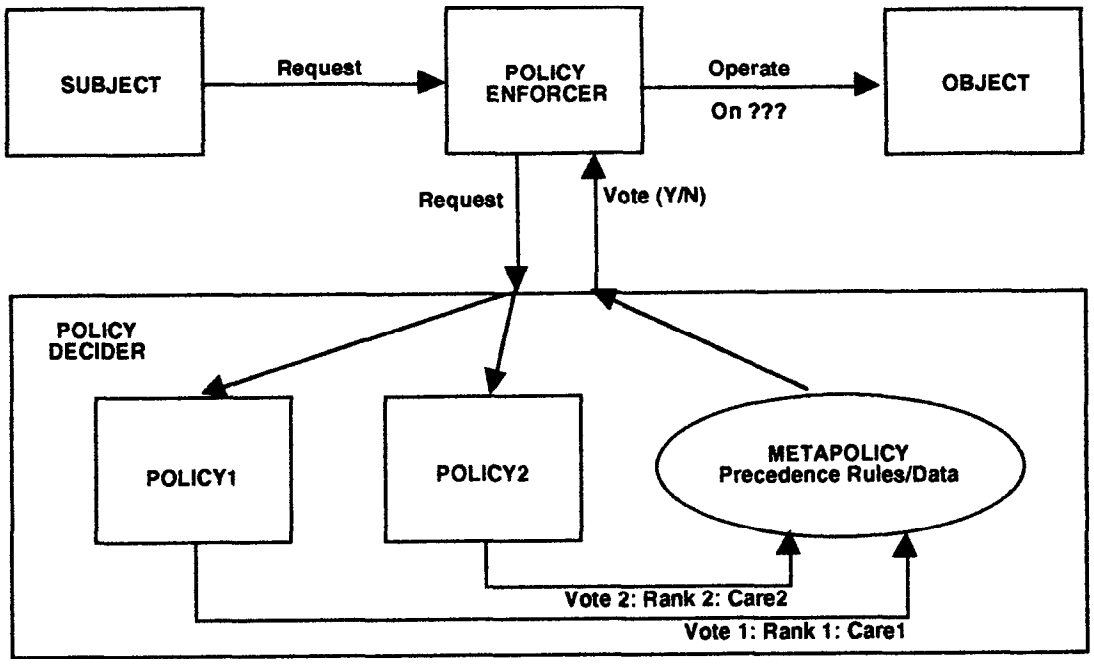
Figure 11. Fuzzy Arrival Times

In fuzzy decision-making, goals, constraints, and decisions can all be fuzzy. The Multipolicy Machine resolves disputes by allowing each policy to assert its rank and "vote" its preference[33] (The vote may be either a "yes", "no", "don't care" or a value representing a point on a continuum) to a conflict-resolution metapolicy. Fred Daum of Raytheon suggested that policies, when voting, also provide a fuzzy number representing how strongly they "feel" about the vote. Illustrated in Figure 12, this "lobbyist" strategy favors policies which "care" more heavily about an outcome.

Marvin Schaefer and others maintain that systems evaluated with their formal models and verification techniques at the TCSEC A1 level provide only an illusion of assurance . Since any element of a formal model can be a fuzzy set, including subjects, objects, and covert channels, fuzzy formal models may offer the promise of more realism. To test this thesis, Professor Sergei Ovchinnikov of San Francisco is building a fuzzy version of the Bell and LaPadula model.

Finally, covert channel bandwidths, degrees of assurance, measures of trust, and many other continua are fuzzy and could be incorporated into the multipolicy paradigm. Assurance for each security policy, for example, might be a set of fuzzy numbers reflecting the thoroughness (e.g. informal model, formal model, prototype) with which the merits of the policy (e.g., consistency, completeness, meeting user needs, architecture, etc.) have been investigated.

Figure 12. Multipolicy Conflict Resolution[34]



Multipolicy Conflict Resolution in Figure 12

1) The 'Subject' wants to operate on the 'Object', but the request must be mediated by the 'Policy Enforcer'.

2) The Policy Enforcer passes the request to the 'Policy Decider' along with the subject and object 'Policy Domain Codes'. The Decider consists of multiple 'Policies' operating in parallel, one for each policy implemented by the system.

3) Based upon policy domain codes, the request is routed to the proper Policies.

4) Using rules and decision data to evaluate the request, each Policy sends its policy precedence 'Ranking', a Vote (e.g., Yes', 'No', 'Don't Care', 'Undecided' or a fuzzy logic number on a continuum), and a fuzzy logic number indicating how much it "Cares" to the Metapolicy.

5) The votes of all the individual policies (Vote 1, Vote 2) in this example) are combined by the Metapolicy and weighed according to its rules as well as the precedence ranking of each policy (Rank 1 and Rank 2 in this example) and how strongly each policy "cares" (Care1, Care2) about its vote.

6) The resulting 'Yes' or 'No' vote is sent back to the Policy Enforcer which then permits or denies the requested operation.

SUMMARY

Fuzzy logic is a relatively new paradigm which may radically impact computer security. It can be used in formal methods, in trusted system analysis and design, in measuring the security of systems, and in representing the imprecise human world of policies and inference. The fuzzy logic paradigm sheds light on many traditional difficulties in computer security, and suggests new directions to follow. The implications are challenging and complicated.

Viewed through the fuzzy logic paradigm, even computer security's clearest concepts, such as the Trusted Computing Base, turn out in practice to be fuzzier and less clear-cut than we supposed. This vagueness is both disturbing and rich and is the rationale for introducing fuzzy set theory, "useful in those complex situations where either some variables are inherently ill-defined or the relationship between many variables is ill-defined".[35]

This paper reviewed basic fuzzy logic concepts, such as crisp and fuzzy sets, prototypes, fuzzy goals, constraints, and decisions, fuzzy logical and mathematical operators, linguistic variables, hedges, and fuzzy voting. It illustrated with simple examples how each of these could be used in the security community, and specifically in the Multipolicy Paradigm. Fuzzy logic can be useful in bridging the human/machine security interface, intrusion detection, modelling non-traditional policies, policy conflict resolution, defining security profiles, and controlling database inference. All of the possibilities mentioned merit additional research.

Although mathematical precision has long been a goal in the security community, computer security is and always has been fuzzy. It is easier to acknowledge this and use tools designed to deal with fuzziness than search in vain for the illusive perfectly secure system.

ACKNOWLEDGEMENTS

Tanya Korelsky, Bill Ford, Bhavani Thuraisingham, and Bill Ricker provided early support for using fuzzy logic in computer security. Professor Sergei Ovchinnikov and Ruth Nelson reviewed this paper and provided helpful comments. John McLean, David Bell and Leonard LaPadula helped by cogently challenging the ideas.

REFERENCES

[1] Hosmer, Hilary H., "The Multipolicy Paradigm", *Proceedings of the 15th National Computer Security Conference*, Baltimore, October, 1992.

[2] Hosmer, Hilary H., "Integrating Security Policies", *Proceedings of the Third RADC Database Security Workshop, June 5-7, 1990, Castile, N.Y.* ed. by Bhavani Thuraisingham, MITRE Technical Report, MTP 385, May 1991.

[3] "Fuzzy logic" is the term coined by Lofti Zadeh for the mathematical insight about imprecision that came to him in 1964. Many have searched for an alternative, more acceptable term, but "fuzzy" is vivid and has stuck.

[4] Zadeh, L.A., "The Concept of a Linguistic Variable", *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, ed. by Yager, Ovchinnikov, R.M. Tong, and H.T. Nguyen, John Wiley and Sons, 1987.

[5] Zadeh, L.A., "The Concept of a Linguistic Variable", *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, ed. by Yager, Ovchinnikov, R.M. Tong, and H.T. Nguyen, John Wiley and Sons, 1987.

[6] Zadeh, Lotfi, "Coping with the Imprecision of the Real World: An Interview with Lotfi A. Zadeh," *Communications of the ACM*, April 1984.

[7] Zimmerman, Hans J., *Fuzzy Sets, Decision-making, and Expert Systems*, Kluwer Academic Publishers, Boston, 1987.

[8] Zadeh, L.A. , *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, ed. by Yager, Ovchinnikov, R.M. Tong, H.T. Nguyen, John Wiley and Sons, 1987.

[9] Ovchinnikov, Sergei, 1993.

[10] Thuraisingham, Bhavani, *Proceedings of the Fifth Annual Computer Security Applications Conference*, Tucson, Arizona, 1989.

[11] Hosmer, Hilary H. "Using Fuzzy Logic to Represent Security Policies in the Multipolicy Paradigm", *ACM SIGSAC Review*, 1993.

[12] Zadeh, Lotfi, "The Concept of a Linguistic Variable-3", *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, ed. by Yager, Ovchinnikov, R.M. Tong, and H.T. Nguyen, John Wiley and Sons, 1987.

[13] Zadeh, Lofti, "A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges", *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh.* ed. by Yager, Ovchinnikov, R.M. Tong, and H.T. Nguyen, John Wiley and Sons, 1987.

[14] Zadeh, "Fuzzy Logic in the Management of Uncertainty", *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh.* ed. by Yager, Ovchinnikov, R.M. Tong, H.T. Nguyen, John Wiley and Sons, 1987.

[15] Rosch, Eleanor (1975), "Cognitive Representations of Semantic Categories", *Journal of Experimental Psychology:General, 104.*

[16] Nelson, Ruth, 1993.

[17] Nelson, Ruth, 1993.

[18] Ovchinnikov, Sergei, 1993

[19] Zadeh, L.A. Bellman R.E. "Decision-making in a Fuzzy Environment", *Management Science*, 1970, 17, 141-164. This paper first introduced fuzzy goals, constraints, and decisions.

[20] Zimmerman, H. J. *Fuzzy Sets, Decision-Making and Expert Systems,* Kluwer Academic Publishers, 1987, p. 73.

[21] Zimmerman, H. J. *Fuzzy Sets, Decision-Making and Expert Systems,* Kluwer Academic Publishers, 1987, p. 73.

[22] Zimmerman, H. J. *Fuzzy Sets, Decision-Making and Expert Systems,* Kluwer Academic Publishers, 1987, p. 73.

[23] Janus Kacprzyk and Andrzej Straszak, "Application of Fuzzy Decision-Making Models for Determining Optimal Policies in 'Stable' Integrated Regional Development", *Fuzzy Sets: Theory and Applications to Policy Analysis and Information Systems, ed. by* P. Wang and S. K. Chang, Plenum Press, New York, 1980.

[24] Zimmerman, H.J. *Fuzzy Sets, Decision-Making and Expert Systems,* Kluwer Academic Publishers, 1987.

[25] Chang, Sheldon S., "On Risk and Decision-Making in a Fuzzy Environment", *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, ed. by Lotfi A. Zadeh, King-Sun Fu, Kokichi Tanaka, and Masamichi Shimura, Academic Press, Inc. New York, 1975.

[26] Asai, Kiyoji, Hideo Tanaka, and Tetsuji Okuda, "Decision-Making and Its Goal in a Fuzzy Environment", *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, ed. by Lotfi A. Zadeh, King-Sun Fu, Kokichi Tanaka, and Masamichi Shimura, Academic Press, Inc. New York, 1975.

[27] Negoita, Constantin, *Expert Systems and Fuzzy Systems,* Benjamin/Cummings Publishing Company, 1985.

[28] Kandel, "Fuzzy Statistics and Policy Analysis", *Fuzzy Sets: Theory and Applications to Policy Analysis and Information Systems*, ed. by P. Wang and S. Chang, Plenum Press, New York, 1980.

[29] Zadeh, L.A., "The Concept of a Linguistic Variable-2", *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, ed. by Yager, Ovchinnikov, R.M. Tong, and H.T. Nguyen, John Wiley and Sons, 1987.

[30] Zimmerman, H. J., *Fuzzy Sets, Decision-Making and Expert Systems,* Kluwer Academic Publishers, 1987, p. 73.

[31] Zadeh, Lofti, *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh.* ed. by Yager, Ovchinnikov, R.M. Tong, and H.T. Nguyen, John Wiley and Sons, 1987.

[32] Hosmer, Hilary H., "Using Fuzzy Logic to Represent Security Policies in the Multipolicy Paradigm", *ACM SIGSAC Review,* 1993.

[33] LaPadula, Leonard, "A Rule-Based Approach to Formal Modeling of a Trusted Computer System", *M91-021*, August 1991.

[34] The diagram and description combine: 1) our visualization of multiple policies operating in parallel with policy conflicts between them resolved by metapolicies; 2) Dr. Marshall Abrams' proposed extended ISO access control policy framework; 3) Leonard LaPadula's voting concept for rule-based systems, and 4) Fred Daum's fuzzy logic "Care About" suggestion.

[35] Jaim, Ramesh, "Fuzzyism and Real World Problems", *Fuzzy Sets: Proceedings of the Symposium on Policy Analysis and Information Systems*, edited by P.P. Wang and S.K. Chang, Durham, North Carolina, 1980.