Fuzzy Sets and Secure Computer Systems

Sergei Ovchinnikov
Mathematics Department
San Francisco State University
1600 Holloway Avenue
San Francisco, CA 94132
sergei@mercury.sfsu.edu

Abstract

The paper presents an attempt to build a mathematical foundation for modeling secure computer systems in a fuzzy environment. It contains a brief tutorial on fuzzy set theory, an introduction to abstract fuzzy systems theory, and offers a fuzzy version of the basic security theorem in the framework of the Bell-LaPadula model.

1 Introduction

The Joint DoD and CIA Security Commission proposes a new security paradigm in which we no longer look for perfect security, but settle for a level of security appropriate to realistic threat estimates. Fuzzy set theory is appropriate to model the new reality because it provides rigorous methods to handle many possible degrees of security.

Formal methods and models are inherent components of the computer security paradigm because they provide for provable security. It is possible to develop formal models for computer security in a fuzzy environment and use fuzzy logic techniques to establish provable security. Not necessarily all components of such models must be fuzzy, but if we want to face the reality of computer security, we have to have at least some fuzziness present in formal models.

The main goal of this paper is to build a mathematical foundation for modeling computer security in a fuzzy environment. In their original publications [1, 2], Bell and LaPadula developed a basic model of a secure computer system based upon general systems theory. We intend to follow their approach and develop a similar model using ideas, language, and techniques of the contemporary fuzzy set theory.

Fuzzy set theory and its branch, fuzzy logic, use a variety of models for basic set operations and logical connectives. All these models employ triangular norms (t-norms) and conorms ([4]) and negation functions as tools for modeling connectives AND, OR and negation NOT. More than often the choice of a particular representation is made ad hoc without any justification. Only very recently such an issue as robustness of fuzzy logics was addressed in [3]. In Section 2 we present an original approach to the theory of t-norms which based on a "parametrization" of a particularly important class of t-norms. This approach will allow for a greater flexibility in our future studies and help verifying secure computers models in a fuzzy environment.

Section 3 presents basic concepts of fuzzy set theory. The main goal of this section is to introduce language and notations that will be uniformly used in these studies. We pay a special attention to fuzzy orderings in this section, because of their potential role in developing Multipolicy Machine Model in a fuzzy environment.

Abstract systems theory is a basis of the Bell-LaPadula model. In Section 4 we present elements of abstract fuzzy systems theory. This section contains mostly original material not found in the pertinent literature.

Finally, in Section 5, the Bell-LaPadula model in a fuzzy environment is described, a fuzzy version of the simple security property is formulated, and the Basic Security Theorem is proven. A totally new approach to trusted computer systems unfolds in this new paradigm. We no longer look for hundred percent secure states but rather consider any state to be secure to some degree. Then a system is considered to be secure if, in its evolution from the initial state, the security levels of states form an nondecreasing sequence.

2 Triangular norms and conorms

Functions

$$M(x,y) = \min\{x,y\},\ M^*(x,y) = \max\{x,y\},$$
and $N(x) = 1 - x$

are standard models for logical connectives AND, OR, and negation NOT in fuzzy set theory. These functions are examples of triangular norms (t-norms), conorms

0-8186-6335-9/94 $04.00

(t-conorms), and negation functions, respectively. The contemporary fuzzy set theory employs a variety of t-norms, t-conorms, and negation functions in modeling logical connectives. In this section we describe particular classes of t-norms, t-conorms, and general negation functions that are most frequently used in applications. An important notion of a residual implication is also introduced.

A *t-norm* T is defined as a function T: $[0,1]^2 \to [0,1]$ satisfying the following properties:
(i) $T(x,1) = x$ *identity*
(ii) $T(x,y) \le T(z,u)$, if $x \le z$ and $y \le u$ *monotonicity*
(iii) $T(x,y) = T(y,x)$ *commutativity*
(iv) $T(x,T(y,z)) = T(T(x,y),z)$ *associativity*
for all $x,y,z,u \in [0,1]$.

Any t-norm T satisfies inequality $T(x,y) \le M(x,y)$ for all $x,y \in [0,1]$. Thus, function M is an extreme case of a t-norm.

An *Archimedean* t-norm is a t-norm satisfying
(v) $T(x,x) < x$
for all x in (0,1). Note, that M is not an Archimedean t-norm. We say that a t-norm T has *zero divisors* if it satisfies
(vi) $T(x,y) = 0$
for some positive x and y.

A 'canonical' example of a t – norm with zero divisors is given by the *Lukasiewicz t – norm*

$$W(x,y) = \max\{x+y-1, 0\}.$$

A 'canonical' example of a t – norm without zero divisors is given by the *product t – norm*

$$\Pi(x,y) = x \cdot y.$$

We call a strictly increasing function ϕ form the unit interval onto itself an *automorphism* of the unit interval. Any automorphism of the unit interval is a continuous function satisfying boundary conditions $\phi(0) = 0$ and $\phi(1) = 1$.

The following two theorems (see [6] and [7]) show that, in a sense, any continuous Archimedean t-norm is 'similar' to one of the two 'canonical' t-norms.

Theorem 1. A t – norm T is a continuous Archimedean t – norm with zero divisors if and only if there exists an automorphism ϕ of the unit interval $[0,1]$, such that

$$T(x,y) = W^{\phi}(x,y) = \phi^{-1}(W(\phi(x),\phi(y))).$$

Function ϕ from Theorem 1 is called a W-generator of T.

Theorem 2. A t – norm T is a continuous Archimedean t – norm without zero divisors if and only if there exists an automorphism ϕ of the unit interval $[0,1]$, such that

$$T(x,y) = \Pi^{\phi}(x,y) = \phi^{-1}(\phi(x)\,\phi(y)).$$

Function ϕ from Theorem 1 is called a Π-generator of T.

A *negation function* N is defined as a strictly decreasing function N: $[0,1] \to [0,1]$ satisfying

$$N(N(x)) = x, \text{ for all } x \in [0,1].$$

Thus defined negation function is a continuous function satisfying boundary conditions $N(0) = 1$and $N(1) = 0$.

A 'canonical' example of a negation is given by $N(x) = 1 - x$.

Theorem 3. ([18]) N is a negation function if and only if there exists an automorphism ϕ of the unit interval, such that

$$N(x) = N^{\phi}(x) = \phi^{-1}(1 - \phi(x)).$$

Function ϕ from Theorem 2 is called a *generator* of N.

A *t-conorm* S is defined as a function S: $[0,1]^2 \to [0,1]$ satisfying the following properties:
(i) $S(0,x) = x$
(ii) $S(x,y) \le S(z,u)$, if $x \le z$ and $y \le u$
(iii) $S(x,y) = S(y,x)$
(iv) $S(x,S(y,z)) = S(S(x,y),z)$
for all $x,y,z,u \in [0,1]$.

Any t-conorm S satisfies inequality $S(x,y) \ge M^*(x,y)$ for all $x,y \in [0,1]$. Thus, function M^* is an extreme case of a t-norm.

Let T and N be a t-norm and negation function, respectively. Then

$$T^*(x,y) = N(T(N(x),N(y))$$

defines a t-conorm and any t-conorm can be represented in this way.

If $T = W$ and $N(x) = 1 - x$, then the corresponding t-conorm W^* is given by

$$W^*(x,y) = \min\{x+y, 1\}.$$

For $T = \Pi$ and $N(x) = 1 - x$ the t-conorm Π^* is given by

$$\Pi^*(x,y) = x + y - xy.$$

For any t-norm T we have $T \le M < M^* \le T^*$. In particular,

$$W < \Pi < M < M^* < \Pi^* < W^*.$$

A triple $< T, T^*, N >$, where T^* is given by the above formula, is called a *De Morgan triple* in fuzzy set theory. Suppose ϕ is a W- or Π-generator of a t-norm T and the same ϕ is a generator of a negation function N. The automorphism ϕ can be regarded as a 'parameter' in our model $< T, T^*, N >$ for logical connectives and a negation function.

Suppose ϕ is a W-generator of T. Then elements of the De Morgan triple have the following representations
$T(x,y) = \phi^{-1}(\max\{\phi(x) + \phi(y) - 1, 0\})$,
$T^*(x,y) = \phi^{-1}(\min\{\phi(x) + \phi(y), 1\})$,
$N(x) = \phi^{-1}(1 - \phi(x))$.

If ϕ is a Π-generator, then we have a different 'parametrization'

$T(x,y) = \phi^{-1}(\phi(x)\,\phi(y))$,
$T^*(x,y) = \phi^{-1}(\phi(x) + \phi(y) - \phi(x)\,\phi(y))$,
$N(x) = \phi^{-1}(1 - \phi(x))$.

By using the above representations, one can construct families of t-norms at will. For instance, for a real parameter p, equations

$T_p(x,y) = (\max(x^p + y^p - 1, 0))^{1/p}$, $p \neq 0$
$T_0(x,y) = \Pi(x,y) = xy$

define a family of t-norms that have no zero divisors if and only if $p \leq 0$.

Let T be a t-norm. Then *R-implication* (where R stands for "residual" [5]) is defined by

$$I(x,y) = \sup\{\, z \mid T(x,z) \leq y \,\}.$$

Theorem 4. Suppose T is a continuous Archimedean t-norm with zero divisors. Then

$$I(x,y) = \phi^{-1}(\min\{1 - \phi(x) + \phi(y), 1\})$$

for all $x, y \in [0,1]$, where ϕ is a W-generator of T. If T has no zero divisors, then

$$I(x,y) = \min\left\{ \frac{\phi(y)}{\phi(x)}, 1 \right\}$$

for all $x, y \in [0,1]$, where ϕ is a Π-generator of T.

It is easy to verify that I satisfies the following conditions:

$I(1,x) = x$,
$I(x,y) = 1$ if and only if $x \leq y$,
$I(x,y) \leq I(x,z)$ if $y \leq z$,
$I(x,y) \geq I(z,y)$ if $x \leq z$.

The approach to the theory of triangular norms presented in this section has been successfully applied to modeling preference relations and collective decision-making in a fuzzy environment (see [6], [7], [8]).

3 Elements of fuzzy set theory

The goal of this section is to introduce terminology and notations. In what follows $\mathbb{I}$ denotes the unit interval $[0,1]$, $x \wedge y = \min\{x,y\}$, and $x \vee y = \max\{x,y\}$.

Let U be a set. A *fuzzy set* A on U is completely defined by its *membership function* $A: U \to \mathbb{I}$. In other words, we do not distinguish between fuzzy sets and their membership functions. The set U is often called the *universe of discourse* or the *domain* of A.

A fuzzy set A is a subset of a fuzzy set B ($A \subseteq B$) iff $A(x) \leq B(x)$ for all $x \in U$. Basic operations of intersection, union, and complement are defined in terms of membership functions as follows

$(A \bigcap B)(x) = A(x) \wedge B(x)$,
$(A \bigcup B)(x) = A(x) \vee B(x)$,
$\bar{A}(x) = 1 - A(x)$

for all $x \in U$. (Occasionally, t-norms, t-conorms, and negation functions are employed in these definitions.)

The set $\mathcal{F}(U) = \mathbb{I}^U$ of all fuzzy sets with domain U is a complete completely distributive lattice. The set $\mathcal{P}(U) = \{0,1\}^U$ of all subsets of U is the maximal Boolean sublattice of $\mathcal{F}(U)$. These sets are called *crisp sets* if one wants to distinguish them from fuzzy sets.

There are different definitions of the difference of two subsets of X in the classical set theory. All of them are equivalent to the definition of the difference in the Boolean algebra $\mathcal{P}(X)$. Since $\mathcal{F}(X)$ is not a Boolean lattice, it is possible to introduce differences between fuzzy sets in a number of different ways. Thus defined differences are not the same in $\mathcal{F}(X)$ but coincide with the standard one when restricted to $\mathcal{P}(X)$.

Here we employ the following definition of the difference in $\mathcal{P}(X)$.

$$B - A = \bigcap \{B' \mid B' \subseteq B,\ B' \cup A = B \cup A\}, \quad (3.1)$$

where A and B are subsets of X. This definition is equivalent to the standard one

$$B - A = \{x \in B \mid x \notin A\},$$

but does not use elements of X explicitly. The main advantage of our definition is that it can be used exactly in the same form in fuzzy set theory. In terms of membership functions, (3.1) can be written in the following form

$$(B - A)(x) = \inf_{x \in X} \{B'(x) \mid B'(x) \leq B(x), \quad (3.2)$$
$$B'(x) \vee A(x) = B(x) \vee A(x)\}.$$

Note, that in the particular case when $B = X$, (3.2) defines so-called *dual intuitionistic negation* [17].

We define operation $\ominus$ on the unit interval $[0,1]$ by

$$\alpha \ominus \beta = \begin{cases} \alpha, & \text{if } \alpha > \beta \\ 0, & \text{if } \alpha \leq \beta \end{cases}$$

Then (3.2) can be written as

$$(B - A)(x) = B(x) \ominus A(x) \quad (3.3)$$

for all $x \in X$.

We shall need two properties of the difference defined by (3.3). First, we have

$$(B - A) \cup A = B \cup A. \quad (3.4)$$

Second,

$$(A - C) \cup (B - C) = (A \cup B) - C. \quad (3.5)$$

Let $U = \prod_{i=1}^{n} U_i$ be a Cartesian product of n sets. A fuzzy set R with the domain U is called a *fuzzy n-ary relation*. In particular, a *fuzzy binary relation on* $X \times Y$ is a fuzzy set with the domain $X \times Y$. Suppose R and S are fuzzy binary relations on $X \times Y$ and $Y \times Z$, respectively. The *composition* of R and S is a fuzzy binary relation on $X \times Z$ is defined by

$$(R \circ S)(x, z) = \sup_{y \in Y} R(x, y) * S(y, z)$$

for all $x \in X, z \in Z$, where $*$ is a binary operation given by a t-norm. By the associativity property of t-norms, thus defined composition is also associative.

Let R be a fuzzy binary relation on $X \times Y$. The *inverse* binary relation R^{-1} is a fuzzy binary relation on $Y \times X$ given by $R^{-1}(x, y) = R(y, x)$. It enjoys usual properties

$$(R \circ S)^{-1} = S^{-1} \circ R^{-1} \quad \text{and} \quad (R^{-1})^{-1} = R.$$

Suppose A and B are fuzzy sets with domains X and Y, respectively, and let R be a fuzzy binary relation on $X \times Y$. The *image* of A under R is a fuzzy set $A \circ R$ with the domain Y defined by

$$(A \circ R)(y) = \sup_{x \in X} A(x) * R(x, y).$$

Similarly, the *inverse image* of B under R is given by

$$(R \circ B)(x) = \sup_{y \in Y} R(x, y) * B(y).$$

If R is a fuzzy binary relation with domain $X \times X$, we say that R is a fuzzy binary relation on X. We define the basic properties of fuzzy binary relations as follows ([16]).

Reflexivity: $R(x, x) = 1, \forall x \in X$.

Irreflexivity: $R(x, x) = 0, \forall x \in X$.

Symmetry: $R(x, y) = R(y, x), \ \forall x, y \in X$.

Asymmetry: $R(x, y) \wedge R(y, x) = 0, \forall x, y \in X$.

Weak Asymmetry: $R(x, y) \wedge R(y, x) < 1, \forall x, y \in X$.

Transitivity: $R(x, z) \geq R(x, y) * R(y, z)$, for all $x, y, z \in X$.

Negative Transitivity: $R(x, z) \leq R(x, y) * R(y, z)$, for all $x, y, z \in X$.

Completeness: $R(x, y) \vee R(y, x) > 0, \forall x, y \in X$.

Strong Completeness: $R(x, y) \vee R(y, x) = 1$, for all $x, y \in X$.

Note that the transitivity property means that $R \circ R \subseteq R$. If R is a reflexive and transitive fuzzy binary relation then $R \circ R = R$.

These and other properties of fuzzy binary relations are used to introduce special classes of binary relations. For instance, a *similarity relation* (fuzzy equivalence relation) is defined as a reflexive, symmetric, and transitive fuzzy binary relation. Let S be a similarity relation on X. A *similarity class* of $a \in X$ is a fuzzy set $S[a]$ defined by $S[a](x) = S(a, x)$. Similarity classes of S for a *fuzzy partition* of X and, conversely, fuzzy partitions generate similarity relations on X (see [16]).

In general, a *fuzzy ordering* is a transitive fuzzy binary relation with some kind of asymmetry property. The following classes of fuzzy orderings play an important role in modeling preference structures. A fuzzy binary relation R on X is a

Partial Ordering if R is asymmetric and transitive,

Weak Ordering if R is asymmetric and negatively transitive,

Linear Ordering if R is a complete weak ordering,

Quasi-Transitive Relation if R is strongly complete and negatively transitive,

Complete Quasi-Ordering if R is strongly complete and transitive,

Reflexive Linear Ordering if R is strongly complete, weakly asymmetric, and transitive.

The following table presents the hierarchy of fuzzy ordering and reflects the *duality* ([16]) between asymmetric and strongly complete fuzzy binary relations.

Partial Orderings	*Quasi-Transitive Relations*
Weak Orderings	*Complete Quasi-Orders*
Linear Orderings	*Reflexive Linear Orderings*

A *fuzzy function* F with domain X and codomain Y is a fuzzy binary relation on $X \times Y$ satisfying

(i) for any $x \in X$ there exists $y \in Y$ such that $F(x, y) = 1$, and

(ii) if $F(x, y') = F(x, y) = 1$, then $y' = y$, for any $x \in X$.

A fuzzy set A on X is said to be *normalized* if there is $x \in X$ such that $A(x) = 1$. It is a *single-peaked* set if there is only one x with this property. The *value* $F(x)$ of a fuzzy function F is the image of a singleton $\{x\}$, i.e., $F(x)$ is a fuzzy subset of Y defined by $F(x)(y) = F(x, y)$. It follows from (i) and (ii) that all values of a fuzzy function are single-peaked fuzzy sets on Y.

A fuzzy function F is *one-to-one* if $F(x', y) = F(x, y) = 1$ implies $x' = x$ for all $y \in Y$. F is *onto* if, for any $y \in Y$, there is $x \in X$ such that $F(x, y) = 1$. The notions of image and inverse image for fuzzy functions are the same as those for fuzzy binary relations.

The concept of *possibility* developed by Zadeh in [9] plays an important role in fuzzy set theory. A very good mathematical treatment of the possibility theory based on measure theory is given in [10]. We use this concept only to give an alternative interpretation for other concepts that are introduced later.

Let Y be a variable taking values in X; then a *possibility distribution*, Π_Y, associated with Y may be viewed as a fuzzy constraint on the values that may assigned to Y. Such a distribution is given by a *possibility distribution function* $F: X \to [0,1]$ which associates with each $x \in X$ the "degree of ease" or the possibility that Y takes x as a value. Obviously, F is just a fuzzy set on X; it is usually assumed that it is a normalized fuzzy set.

4 Elements of abstract fuzzy systems

Although some authors, including Lotfi Zadeh, have investigated general systems in a fuzzy environment (see, for instance, [11] and [12]), "... there is virtually no work done on mathematical theory of general fuzzy systems" (George Klir, personal communication.) Since we attempt to develop an approach to computer security in a fuzzy environment similar to the Bell-LaPadula model which is based on mathematical general systems theory, we present in this section elements of abstract fuzzy systems theory including such important notions as fuzzy input, output, and fuzzy state. Basically, we follow here the ideas presented in [13] and [14] and begin with introducing basic notions in the nonfuzzy case.

Suppose X and Y are two abstract sets which are usually considered as inputs and outputs of the system. An *abstract (terminal) system* is a proper binary relation $S \subset X \times Y$. We use the same symbol S for the characteristic function of this binary relation. For a given $x \in X$ the set of all $y \in Y$ such that xSy (or, equivalently, $S(x,y) = 1$) is the set of all possible outputs for a given input x. Boolean valued function $S(x,y)$ may be viewed then as a crisp possibility distribution on outputs. It is assumed that the domain of S is X, i.e., for any $x \in X$ there exists $y \in Y$ such that xSy. If C is a set and function $R: C \times X \to Y$ satisfies

$$xSx \Leftrightarrow (\exists c)[R(c,x) = y],$$

we say that R is a *global response function* and C is a *state object*.

The notion of state is very important in systems theory. In traditional approaches this notion plays a primary role along with input/output sets and various auxiliary functions. The abstract systems theory "... starts from the input/output pairs ... and derives the concept of state as a secondary concept." ([14]) The following is Theorem 1.1 in [13].

Proposition. For each abstract system $S \subset X \times Y$ there exists a global response function R.

For each given $c \in C$, function R may be viewed as a binary relation R_c on $X \times Y$. Then the condition defining global response function can be written in the following form

$$S = \bigcup_{c \in C} R_c$$

In the framework of fuzzy set theory we develop the following approach to abstract fuzzy systems theory.

An *abstract fuzzy system* is a fuzzy binary relation S on $X \times Y$. We assume that the domain, $S \circ Y$, of S is X, i.e., for any $x \in X$ there is $y \in Y$ such that $S(x,y) = 1$. A *fuzzy input (fuzzy output)* is a fuzzy subset of X (Y). We shall assume that fuzzy inputs and outputs are normalized fuzzy sets. The image O of a fuzzy input I under S may be viewed as a fuzzy set of all possible outputs corresponding to I. In our notations

$O = I \circ S$ or, equivalently, $O(y) = \sup_{x \in X} I(x) * S(x,y)$.

In particular, for $I = \{a\}$, $O(y) = S(a,y)$. This function defines the possibility distribution of outcomes corresponding to the (crisp) income a.

Let C be a set and R be a fuzzy function from $C \times X$ to Y. We say that R is a *fuzzy global response function* for a fuzzy system S if

$$S(x,y) = \sup_{c \in C} R(c,x,y) \qquad (*)$$

for all $x \in X, y \in Y$. The set C is called *fuzzy state object*. For any given $c \in C$, R defines a fuzzy function $R_c: X \to Y$. Then

$$S = \bigcup_{c \in C} R_c$$

Thus our definition is a fuzzy analog of the crisp one.

The following theorem asserts the existence of a fuzzy global response function for any fuzzy abstract systems S.

Theorem. For any fuzzy abstract system there exists a fuzzy global response function.

Proof. Let $S \subset X \times Y$ be a fuzzy system. Consider

$$C = \{c \mid c \subseteq S \text{ and } c \text{ is a fuzzy function}\}$$

and define $R(c,x,y) = c(x,y)$. It follows from the rest of the proof that $C \neq \emptyset$.

Thus defined R is a fuzzy function. Indeed, for any given pair (c,x), there is a unique $y \in Y$ such that $R(c,x,y) = c(x,y) = 1$, since c is a fuzzy function.

To prove $(*)$, we note first that

$$R(c,x,y) = c(x,y) \leq S(x,y).$$

Therefore

$$S(x,y) \leq \sup_{c \in C} R(c,x,y).$$

We prove now that, for any given x and y, there is c such that

$$S(x,y) = R(c,x,y).$$

Consider two cases.

(1) $S(x,y) = 1$. Since the domain of S is X, for any $u \in X$ there exists $v \in Y$ such that $S(u,v) = 1$. Consider crisp function c such that $c(x) = y$ and $c(u) = v$ for

$u \neq x$, where v is any element in Y such that $S(u,v) = 1$. Obviously, $c \in C$ and

$$R(c,x,y) = c(x,y) = 1 = S(x,y).$$

(2) $S(x,y) < 1$. Like in the previous case, one can find a crisp function $c \in C$. We define $c^*(x,y) = S(x,y)$ and $c^*(u,v) = c(u,v)$ otherwise. Thus defined c^* is a fuzzy function since $S(x,y) < 1$. Obviously, $c^* \in C$.

□

Instead of developing a theory of fuzzy abstract *time* systems in the most general way, we present here an approach based on more traditional ideas. Namely, we shall use notions of state transition and output functions to describe the dynamics of fuzzy systems.

Let $T = \{1, 2, \ldots, t, \ldots\}$ be the *time set*, A and B *input and output alphabets*, $X = A^T$ and $Y = B^T$ input and output sets, respectively, C an abstract set that we shall call a *state space*. Suppose also that functions $\rho: C \times A \to B$ and $\phi: C \times A \to C$ are given. These functions are called *output* and *state-transition* functions respectively. Equations

$$y_t = \rho(c_t, x_t) \text{ and } c_{t+1} = \phi(c_t, x_t)$$

for all $t \in T$, are called *state equations* of the system.

The next step is to assume that, for any given $t \in T$, x_t, y_t, and c_t are fuzzy sets on A, B, and C, respectively, and functions ρ and ϕ are fuzzy relations on $C \times A \times B$ and $C \times A \times C$, respectively. Then *fuzzy state equations* of a *fuzzy system* are now

$$\sup_{\substack{c \in C \\ a \in A}} c_t(c) * x_t(a) * \rho(c,a,b)$$

and

$$c_{t+1}(c) = \sup_{\substack{d \in C \\ a \in A}} c_t(d) * x_t(a) * \phi(d,a,c)$$

or, more compactly,

$$y_t = c_t \circ x_t \circ \rho \text{ and } c_{t+1} = c_t \circ x_t \circ \phi$$

for all $t \in T$.

One can view a fuzzy state as a possibility distribution over C, i.e., the actual state is one of the elements of C; but since the process behavior is partly unknown, several states are possible with a non zero possibility degree. ρ and ϕ can be then viewed as *conditional* possibility distributions. For instance, $\phi(c_{t+1}, x_t, c_t)$ is the possibility for the state to be c_{t+1} at time $t+1$, knowing that the state and the input at time t are c_t and x_t, respectively. In some situations relations ρ and ϕ can be directly obtained through a linguistic description using names of fuzzy sets on C, involved in fuzzy conditional propositions. Then state equations can be established using the above formulas. Such a fuzzy model corresponds to an approximate (linguistic) description of a complex system whose equations are possibly unknown.

Consider now the case of nonfuzzy inputs x_t. Then the state equations can be written in the following form

$$y_t = c_t \circ \rho_{x_t} \text{ and } c_{t+1} = c_t \circ \phi_{x_t}$$

where ρ_{x_t} and ϕ_{x_t} are fuzzy output and state-transition functions when the input is x_t. Expanding the last formula, we have

$$c_{t+1} = c_0 \circ \phi_{x_0} \circ \phi_{x_1} \circ \ldots \circ \phi_{x_t} = c_0 \circ \Phi_{\bar{x}_t}$$

where $\bar{x}_t = x_0 x_1 \ldots x_t$ is an input string. The output equation can be now written in this form

$$y_{t+1} = c_0 \circ \Phi_{\bar{x}_t} \circ \rho_{x_{t+1}}.$$

To illustrate our approach we consider the following simple example (cf. [14], section 3.1.1.) Suppose we have a vending machine which accepts quarters as inputs and gives one fifty-cent can of soft drink as the output. This system has two states. The first state $c_\emptyset$ is when no quarter has been put in yet. The second state c_q corresponds to the case when a quarter has been put in beforehand. Thus the state space is $C = \{c_\emptyset, c_q\}$. The input alphabet is a singleton $A = \{q\}$ where symbol q represents "putting a quarter in the machine." Symbols λ and χ are elements of the output alphabet B representing the events "nothing in the output" and "a can of soft drink at the output", respectively. Since the input set is a singleton, the output and state-transition functions are just fuzzy binary relations and can be described by 2×2 matrices. We define them as follows:

$$\rho = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \text{ and } \phi = \begin{bmatrix} \alpha & 1 \\ 1 & \alpha \end{bmatrix}.$$

where $0 < \alpha < 1$. The "fuzziness" of our machine is characterized by the possibility α that the machine does not accept the input (it either rejects the quarter or "swallows" it.) For instance, $\rho_{21} = \alpha$ means that the possibility of outcome λ (nothing out) in the state c_q (a quarter is in the machine) is α while $\rho_{22} = 1$ means that in the same state the possibility of getting a can of soft drink is 1. Since the state space and the output alphabet are two-element sets, we represent fuzzy states and fuzzy outputs by two-dimensional vectors. Simple calculations show that there are two possible cases:

1) $t = 2k$.

Then $c_{t+1} = < 1, \alpha >$ and $y_{t+1} = < 1, \alpha >$;

2) $t = 2k+1$.

Then $c_{t+1} = < \alpha, 1 >$ and $y_{t+1} = < \alpha, 1 >$.

For instance, in the first case, the machine is in the state $c_\emptyset$ with possibility 1 and in the state c_q with possibility α. Similarly, in the same case, the possibility of getting a can of soft drink is α while the possibility of getting no output is 1.

We conclude this section with a citation from [15]. "A general theory of fuzzy systems perhaps demands more imagination than a straightforward extension of classical concepts of nonfuzzy system theory. Since the theory of approximate reasoning, initiated by Zadeh, radically departs from multivalent logics, a theory of fuzzy systems should perhaps be developed outside of the conceptual framework of classical system theory."

5 The Bell-LaPadula model in a fuzzy environment

First, we introduce in this section elements of the Bell-LaPadula (BLP) model [1] and then describe a fuzzy system based on these elements.

$\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$. Subjects; processes and domains. Although each subject is not fuzzy, it is plausible to consider a fuzzy set of subjects assigning to each subject the degree to which this subject can, say, affect the system state.

$\mathcal{O} = \{O_1, O_2, \ldots, O_m\}$. Objects; files, terminals, programs, devices. Each subject is also considered as an object. A fuzzy set on O would, for example, define the degree to which a given subject is considered to be an object.

$\mathcal{C} = \{C_1, C_2, \ldots, C_q\}$. $C_1 > C_2 > \ldots > C_q$. Classifications; elements of this set are clearance levels of subjects or classifications of objects. Classifications could be values of a linguistic variable and thus represented by fuzzy sets. In this case, the ordering of classifications is also fuzzy.

$\mathcal{K} = \{K_1, K_2, \ldots, K_r\}$. Need-to-know categories; project numbers, access privileges. Linguistic variables could be used to describe elements of this set.

$\mathcal{A} = \{A_1, A_2, \ldots, A_p\}$. Access attributes; read, write, copy, append, owner, control. At least some of the attributes could be values of linguistic variables.

$\mathcal{R} = \{R_1, R_2, \ldots, R_u\}$. Requests; inputs, commands, requests for access to objects by subjects.

$\mathcal{X} = \mathcal{R}^T$. Request sequences; a typical element is $x = \{x_t\}$.

$\mathcal{D} = \{D_1, D_2, \ldots, D_v\}$. Decisions; outputs, answers, "yes", "no", "error". Elements of this set are very likely described in terms of linguistic variables.

$\mathcal{Y} = \mathcal{D}^T$. Decision sequences; a typical element is $y = \{y_t\}$.

$\mathcal{F} = \mathcal{C}^{\mathcal{S}} \times \mathcal{C}^{\mathcal{O}} \times (\mathcal{P}(\mathcal{K}))^{\mathcal{S}} \times (\mathcal{P}(\mathcal{K}))^{\mathcal{O}}$. Classification/need-to-know vectors. An arbitrary element of F is written $\overline{f} = (f_1, f_2, f_3, f_4)$. f_1: subject-classification function; f_2: object-classification function; f_3: subject need-to-know function; f_4: object need-to-know function. In a fuzzy environment, these functions are fuzzy functions and the power set $\mathcal{P}(K)$ is substituted by the set of all fuzzy sets on $\mathcal{K}$.

$\mathcal{M} = \{M_1, M_2, \ldots, M_{nm2^p}\}$. The set of all possible access matrices. The (i, j) entry of an access matrix is an element of $P(A)$ indicating S_i's access attributes relative to O_j.

$\mathcal{V} = \mathcal{P}(\mathcal{S} \times \mathcal{O}) \times \mathcal{M} \times \mathcal{F}$. States.

$\mathcal{Z} = \mathcal{V}^T$. State sequences. z_t is the t-th state in the state sequence $z \in \mathcal{Z}$.

Although it is possible to fuzzify most of the elements of the standard BLP model, we are concerned only with fuzzy states because they are used in the definition of the simple security property and the basic security theorem. In addition, in our simplified fuzzy version of the BLP model, the state-transition relation W is assumed to be a crisp relation.f

In the BLP model, a state v is a triple $(b, M, \overline{f})$ where $b \subseteq \mathcal{S} \times \mathcal{O}$ indicating which subject have access to which objects in the state v, M is the access matrix in the state v, and $\overline{f} = (f_1, f_2, f_3, f_4)$ is the objet/subject classification/need-to-know vector in the state v.

In a fuzzy environment, b is a fuzzy subset of $\mathcal{S} \times \mathcal{O}$. In other words, b is a binary relation on $\mathcal{S} \times \mathcal{O}$ and $b(s, o)$ can be interpreted as the degree to which subject s has an access to object o.

To model $\overline{f}$ using fuzzy sets, we first reformulate the simple security condition. In the standard BLP model, $(s, o) \in \mathcal{S} \times \mathcal{O}$ satisfies the *security condition relative to* $\overline{f}$ if

$$f_1(s) \geq f_2(o) \tag{5.1}$$

and

$$f_3(s) \supseteq f_4(o). \tag{5.2}$$

This first inequality defines a binary relation $f' \subseteq \mathcal{S} \times \mathcal{O}$

$$(s, o) \in f' \Leftrightarrow f_1(s) \geq f_2(o).$$

Similarly, the second inequality defines a binary relation $f'' \subseteq S \times O$

$$(s, o) \in f'' \Leftrightarrow f_3(s) \supseteq f_4(o).$$

Let $f = f' \cap f''$. Then (s, o) satisfies the security condition relative to $\overline{f}$ if and only if $(s, o) \in f$. In terms of relations b and f, the standard simple security property can be written in the following form

$$b \subseteq f. \tag{5.3}$$

To introduce a fuzzy version of this condition, we need a more formal definition of binary relations f' and f''. First, consider the following diagram

$$\begin{array}{ccccc} & & f_1 & & \\ & S & \rightarrow & C & \\ f' & \downarrow & & \downarrow & L \\ & O & \rightarrow & C & \\ & & f_2 & & \end{array}$$

where $L =$ " $\geq$ " is the linear ordering on C. Then, by the definition of f', $f' = f_1 \circ L \circ f_2^{-1}$, where f_2^{-1} is the inverse relation, not the inverse function.

In a fuzzy environment, we assume that L is a fuzzy ordering on C and define $f_1 : S \rightarrow C$ and $f_2 : O \rightarrow C$ as fuzzy functions. Then, for instance, $f_1(s, c)$ is the degree to which subject s has classification c. By definition of a fuzzy function, for any subject s, there is exactly one classification c such that $f_1(s, c) = 1$. The same is true for f_2. Fuzzy functions f_1 and f_2 together with fuzzy order L on C define a fuzzy binary relation f' on $S \times O$ as $f' = f_1 \circ L \circ f_2^{-1}$ or, equivalently,

$$f'(s, o) = \bigvee_{c,c' \in C} f_1(s, c) * L(c, c') * f_2(o, c')$$

where $*$ is a t-norm (in particular, $* = \wedge = \min$). Then $f'(s, o)$ may be interpreted as the degree to which s's clearance is higher then o's classification. In the particular case when both functions f_1 and f_2 are crisp, $f'(s, o) = 1$ if and only if (s, o) satisfies simple security condition relative to f.

We treat the second part of the simple security property (5.2) in a similar manner. In this case, $L =$ " $\supseteq$ " is inclusion relation on $\mathcal{F}(K)$ and f_3 and f_4 are fuzzy functions from S and O to $\mathcal{F}(K)$. Then a fuzzy binary relation f'' on $S \times O$ is defined by

$$f''(s, o) = \bigvee_{A \supseteq B} f_3(s, A) * f_4(o, B)$$

where $A, B \in \mathcal{F}(K)$.

In a fuzzy environment, all states are assumed to be secure to some degree. If condition (5.3) is satisfied for fuzzy b and f, we say that the state is *secure to the degree 1*, or, simply, *secure*. In other words, the state is secure if, for any pair (s, o), the degree to which s has access to o does not exceed the degree to which the clearance of s is greater than the classification of o

$$b(s, o) \leq f'(s, o)$$

and the degree to which s has access to o does not exceed the degree to which need-to-know categories of s contain need-to-know categories of o

$$b(s, o) \leq f''(s, o).$$

In general, the *security level* of the state (b, M, f) should measure the degree to which b is a subset of f. Consider the *height* of the difference $b - f$

$$h(b - f) = \sup_{(s,o)}((b - f)(s, o)) = \sup_{(s,o)}(b(s, o) \ominus f(s, o)).$$

This number is a natural measure of the degree to which b is *not* a subset of f. Then we define the security level of a state as follows.

Simple security property. The *security level* σ_v of the state $v = (b, M, f)$ is given by

$$\sigma_v = 1 - h(b - f) = 1 - \sup_{(s,o)}(b(s, o) \ominus f(s, o)) \tag{5.4}$$

Since σ_v does not depend on M, we shall also use notation $\sigma(b, f)$ for σ_v.

Consider two extreme cases. First, suppose $\sigma_v = 1$. Then $b(s, o) \ominus f(s, o) = 0$, or equivalently, $b(s, o) \leq f(s, o)$. This is the case of a secure state. The converse is obviously also true. Thus $\sigma_v = 1$ if and only if the state v is secure. Suppose now that $\sigma_v = 0$. Then there exists a pair (s, o) such that $b(s, o) = 1$ and $f(s, o) < 1$. In other words, s has total access to o, but the degrees to which clearance of s is greater than classification of o and need-to-know categories of s contain need-to-know categories of o are less than one. We consider such a state v as totally insecure.

Let $T = \{1, 2, \ldots, t, \ldots\}$ be the time set. Following [1], we denote $Z = V^T$ the set of all state sequences. An arbitrary element of Z is written $z = \{z_1, z_2, \ldots, z_t, \ldots\}$. In addition, z_0 denotes a specified initial state. Let $X = R^T$ and $Y = D^T$ be the sets of all request and decision sequences, respectively. For a given state-transition relation

$$W \subseteq R \times D \times V \times V,$$

the system $\Sigma(R, D, W, z_0) \subseteq X \times Y \times Z$ is defined by

$$(x, y, z) \in \Sigma(R, D, W, z_0) \Leftrightarrow (x_t, y_t, z_t, z_{t-1}) \in W,$$

for all $t \in T$.

In a fuzzy environment, we define the security level σ_z of the state sequence z by

$$\sigma_z = \inf_{t \in T}\{\sigma_{z_t}\}.$$

Secure systems. $\Sigma(R, D, W, z_0)$ is a *secure system* if, for any

$$(x, y, z) \in \Sigma(R, D, W, z_0),$$

the security level of z is not less than the security level of the initial state z_0.

Thus, the system is secure if $\sigma_{z_t} \geq \sigma_{z_0}$ for all $t \in T$. In other words, at any moment t, the security level of the state z_t in a secure system is not less than the security level of the initial state z_0.

We need the following technical result to establish the main theorem of this section.

Lemma. Let $v = (b, M, f)$ and $v^* = (b^*, M^*, f)$ be two states with the same f. Then

$$\sigma(b^*, f) \geq \sigma(b, f) \wedge \sigma(b^* - b, f). \tag{5.5}$$

Proof. We have, by (3.4) and (3.5),

$$[(b^* - b) - f] \cup (b - f) = (b^* \cup b) - f \supseteq b^* - f.$$

Therefore,

$$h((b^* - b) - f) \vee h(b - f) \geq h(b^* - f),$$

or, equivalently,

$$[1 - h((b^* - b) - f)] \wedge [1 - h(b - f)] \leq 1 - h(b^* - f).$$

By (7),

$$\sigma(b^*, f) \geq \sigma(b, f) \wedge \sigma(b^* - b, f). \qquad \square$$

Basic Security Theorem. Suppose that $(r, d, (b^*, M^*, f^*), (b, M, f)) \in W$ implies

(i) $f^* = f$

(ii) $\sigma(b^* - b, f^*) \geq \sigma(b, f)$.

Then $\Sigma(R, D, W, z_0)$ is secure.

Proof. It suffices to prove that the sequence $\{\sigma_{z_t}\}_{t \in T}$ is nondecreasing. For a given $t \in T$, let $z_{t-1} = (b, M, f)$ and $z_t = (b^*, M^*, f^*)$ be two consecutive states. By (i), $f^* = f$, and, by (5.5) and (ii), $\sigma(b^*, f) \geq \sigma(b, f)$. Thus $\sigma_{z_t} \geq \sigma_{z_{t-1}}$. $\square$

6 Conclusion

To make this paper self-content, we first outlined basics of the theory of triangular norms, fuzzy set theory, and presented some elements of fuzzy abstract systems theory. Thus we defined language and introduced notations that are used in our ongoing work on modeling computer security in a fuzzy environment.

Just recently, a new computer security paradigm has been proposed which no longer looks for a perfect security but rather intents to use risk management techniques to handle security levels appropriate for particular tasks. Fuzzy set theory can be used in developing such risk management techniques.

In this paper, we have shown that it is possible to have provable security in a fuzzy environment. Our version of the Bell-LaPadula model considers any state of the system as secure to some degree. Then a system is secure if these degrees form a nondecreasing sequence on consecutive states. A fuzzy version of the Basic Security Theorem has been proven establishing a necessary condition for a system to be secure in a fuzzy environment.

Acknowledgment

Work performed under subcontract DSI-4300-O, Contract F19628-C-002.

References

[1] D.E. Bell and L.J. LaPadula, Secure computer systems: mathematical foundations, ESD-TR-73-278, vol. 1, ESD/AFSC, Hansom AFB, Bedford, Mass., Nov. 1973.

[2] D.E. Bell and L.J. LaPadula, Secure computer systems: a mathematical model, ESD-TR-73-278, vol. 2, ESD/AFSC, Hansom AFB, Bedford, Mass., Nov. 1973.

[3] H.T. Nguen,V. Kreinovich, D. Tolberg, On robustness of fuzzy logics, in: Proc. of the 2nd IEEE Int. Conference on Fuzzy Systems, San Francisco, March 28 - April 1, 1993, 543-547.

[4] B. Schweizer and A. Sklar, *Probabilistic Metric Spaces* (North-Holland, 1983).

[5] Dubois, D. and Prade, H., Fuzzy sets in approximate reasoning I, Fuzzy Sets and Systems **40** (1991) 143-202.

[6] S. Ovchinnikov, On fuzzy preference relations, Int. J. of Intelligent Systems **6** (1991), 225-234.

[7] S. Ovchinnikov and M. Roubens, On strict preference relations, Fuzzy Sets and Systems **43** (1991) 319-326.

[8] S. Ovchinnikov, Social choice and Lukasiewicz logic, Fuzzy Sets and Systems **43** (1991) 275-289.

[9] L. Zadeh, Fuzzy sets as a basis for possibility theory, Fuzzy Sets and Systems **1** (1978), 3-28.

[10] G. Klir and T. Folger, *Fuzzy Sets, Uncertainty, and Information* (PrenticeHall, 1988)

[11] L.A. Zadeh, Toward a theory of fuzzy systems, in: Aspects of Network andSystem Theory (R.E. Kalman and N. De Carlis, eds., 1971) 469-490.

[12] C.V. Negoita, *Fuzzy Systems* (Abacus Press, 1981).

[13] M.D. Mesarovic and Y. Takahara, *General Systems Theory: MathematicalFoundations* (Academic Press, 1975).

[14] M.D. Mesarovic and Y. Takahara, *Abstract Systems Theory* (Lecture Notes in Control and Information Sciences, v. 116, Springer-Verlag, 1989)

[15] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications* (Academic Press, 1980).

[16] S. Ovchinnikov, Similarity relations, fuzzy partitions, and fuzzy orderings, Fuzzy Sets and Systems **40** (1991) 107-126.

[17] S. Ovchinnikov, General negations in fuzzy set theory, J. of Math. Anal. and Appl. **92** (1983) 234-239.

[18] E. Trillas, Sobre functiones de negacion en la theoria de conjuntos difusos, Stochastica **III** (1979), 47-60.