Risk-based Access Control Systems Built on Fuzzy Inferences

Qun Ni
Computer Science Dept
Purdue University
W. Lafayette, IN 47906, USA
ni@cs.purdue.edu

Elisa Bertino
Computer Science Dept
Purdue University
W. Lafayette, IN 47906, USA
bertino@cs.purdue.edu

Jorge Lobo
IBM T. J. Watson
Research Center
Hawthorne, NY 10532, USA
jlobo@us.ibm.com

ABSTRACT

Fuzzy inference is a promising approach to implement risk-based access control systems. However, its application to access control raises some novel problems that have not been yet investigated. First, because there are many different fuzzy operations, one must choose the fuzzy operations that best address security requirements. Second, risk-based access control, though it improves information flow and better addresses requirements from critical organizations, may result in damages by malicious users before mitigating steps are taken. Third, the scalability of a fuzzy inference-based access control system is questionable. The time required by a fuzzy inference engine to estimate risks may be quite high especially when there are tens of parameters and hundreds of fuzzy rules. However, an access control system may need to serve hundreds or thousands of users. In this paper, we investigate these issues and present our solutions or answers to them.

Categories and Subject Descriptors

C.2.0 [**Computer Communication Networks**]: General—*security and protection*; D.4.6 [**Operating Systems**]: Security and Protection—*Access Controls*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Management, Security, Standardization

Keywords

Risk, Access Control, Fuzzy Inference

1. INTRODUCTION

The inflexibility of binary decisions typically taken by current access control systems is a major inhibitor to information sharing when dealing with events in critical organizations, e.g. hospitals, intelligence departments, fire departments, and military [13]. Specifically, these systems are not able to meet the requirements of these organizations in a time-efficient manner. For example, when there is an emergency in a sensitive building, such as a terrorist attack, professionals, such as firemen, may not receive in time the clearance to obtain the building information. We thus need access control systems able to grant accesses to subjects based on the assessment of the current situation and possible risks, even when subjects lack proper permissions. Risk-based access control systems (RAC) have been recently proposed [13, 8] to address such a need.

The main idea of RAC is that access requests from risky subjects, for example firemen without clearance, can be allowed given that some mitigating actions, referred to as *post-obligations*, will be taken after the event in order to minimize the possibility of a future information leak, for example, requiring the firemen to sign non-disclosure agreements and performing a background check on involved firemen.

Clearly, the critical step of RAC is the risk estimation of an access request that consists of the possibility of an information leak in the future and the value of the information. Whether an access request should be allowed and what kind of mitigation should be adopted purely depend on the risk estimation. Unfortunately, risk estimation has proven to be a challenging task [20] due to various reasons. One goal of risk estimation is to predicate the *future* possibility of information disclosure resulting from the current access. Determining such a possibility is inherently hard.

Even worse, such an estimation has often to rely on incomplete or imprecise information and knowledge about relevant risk factors, e.g. subjects' background. The lack of such information in practice precludes approaches purely based on machine learning techniques or probability theory, e.g. Bayesian networks. The evaluation of the value of information, another goal of the risk estimation, has similar difficulties.

However, we believe that a fuzzy inference system is a good candidate to support the estimation of access risks. A fuzzy inference system, detailed in Section 2, is a mathematically sound approach [17] for inferring an unambiguous consequence from vague evidences and subjective if-then rules. There are some good reasons to support our belief.

First, we usually have good sources for *subjective* knowledge about risk factors and rules to estimate access risks [3]. One source is the past experience, e.g. administrators and security managers usually have some personal experiences [1] about risk factors and rules from best practices. Another source is collective knowledge. It is usually the case that chief security officers, system administrators, and security researchers share high level experience and possibly effective rules without disclosing the details. This kind of information is sufficient for security officers to come up with their own concrete rules based on organizational requirements. More-

[1]A typical requirement of a job position of security administrators is at least 2-3 years past experience in a similar job position.

ASIACCS'10 April 13–16, 2010, Beijing, China.

over, this kind of subjective knowledge can be naturally translated into the rule base in a fuzzy inference system.

Second, subjective knowledge usually contains some vague concepts, e.g. if the possibility of a terrorist attack to the Pentagon is very high and the confidence that firemen come from nearby fire departments is high, then the risk incurred by letting these firemen access the building map of the Pentagon is low. These vague concepts, e.g. high and low, can be described naturally by carefully defined membership functions in a fuzzy inference system.

Third, fuzzy inference systems have well-studied semantics [22] and thousands of successful applications [24] in engineering, medicine, management, decision support, and psychology, from space shuttles to home appliances. More importantly, fuzzy inference rules can approximate arbitrary functions with unlimited accuracy [17]. This means that the *actual* function of access risk estimation, if existing, can always be approximated by fuzzy rules and membership functions.

Last but not least, fuzzy inference systems are able to combining both subjective knowledge and objective evidences. The accuracy of fuzzy rules and membership functions can be further improved by applying machine learning techniques once we have sufficient objective information about access risk estimation [10, 4]. Moreover, given sufficient training samples, a fuzzy inference system, including its membership functions and fuzzy rules, can be built automatically to precisely match given data [21].

Unfortunately, the application of fuzzy inference systems to estimate access risks raises some issues. The answers to these issues are the focus of this paper. First, as indicated by [15, 9], there are multiple operations in each fuzzy operation category, such as conjunctive operations and disjunctive operations in fuzzy rules, and aggregation operations in the rule consequences. When different operations, e.g. different conjunctions, are selected for calculating risk factors, different risk estimations may be generated. A natural question would be "which operation is the most appropriate one?" We provide criteria to help users in determining the best operation.

Second, risk-based access control systems, though they may improve information flow and better address requirements from critical organizations, may result in damages by malicious users before the mitigating actions take effect. The reason is that there is often a non-trivial time window between the accesses and the execution of the corresponding obligations. Malicious users may take advantage of this window to access pieces of valuable information. We provide an effective method to ensure that the overall damage in the worst case is still under control.

Third, the scalability of a fuzzy inference-based access control system seems to be questionable. Fuzzy inference systems need a non-trivial time to estimate risks especially when there are tens of parameters and hundreds of fuzzy rules. However, an access control system may need to serve hundreds or thousands of users. Therefore a fuzzy inference-based access control system might be too computationally expensive. In this paper, we verify the efficiency of a fuzzy inference system in various combinations of different inference settings and difference access control settings.

The rest of this paper is organized as follows: Section 2 presents a fuzzy BLP example and compares it with an existing Fuzzy MLS [8]. Section 3 discusses the algebraic properties of different operations and defines measures to choose the operations. Relevant theorems to support these choices are presented. Section 4 investigates how to control the overall damage and proposes our solution. Section 5 investigates the scalability problem of fuzzy inference by experiments. Section 6 discusses related work. Section 7 concludes the paper and suggests a future direction.

2. MOTIVATION EXAMPLE

The recently proposed Fuzzy Multi-Level Security (Fuzzy MLS) [8] is an example of risk-based access control systems[2]. Fuzzy MLS quantifies the risk of an access request based on a sigmoid function on the difference between a subject security label and an object security label. The larger the difference is, the higher the risk is. The result is described as a real number in the interval $[0, 1]$, where 1 represents an absolute deny (the highest risk), and 0 represents an absolute permit (the lowest risk). Fuzzy MLS further divides the interval $[0, 1]$ into n sub-intervals between 0 and 1, referred to as risk bands. If the risk of an access request is evaluated to a band, the request is allowed only if the risk mitigation measures associated with the band are applied. Mitigation measures might be obligations that require some subject to perform some operations, e.g., obtain confirmation from a direct manager, either before an access authorization or after the access authorization.

In this section, we apply a fuzzy inference system to develop a different risk-based access control example and compare these two approaches. Both examples are based on the same risk factors, subject security labels, and object security labels.

2.1 A Fuzzy BLP Example

Suppose that there are some documents and an automatic document sensitivity score system. The score system, like a FICO credit score [1], calculates a document score based on four categories of document, each of which has an upper score bound: authors (300), contents (300), departments (200), intended audiences (200). Each category contains some mandatory and optional properties that have different values. Each value has a sensitivity score defined by security experts. The score of missing properties (no value in that property) is ignored. Given a document, its score is calculated by the sum of its property scores. The lowest document score (sum of the lowest scores of mandatory property values) is 500 and the highest score is 1000.

The security labels of documents are divided into unclassified, classified, secret, and top secret based on their scores. If we adopt the following crisp boundary scores for security labels: 500-600 (unclassified), 601-750 (classified), 751-900 (secret), 901-1000 (top secret), we may feel that the sensitivity of a document with score 601 (classified) might be overestimated while the sensitivity of a document with score 600 (unclassified) is underestimated. To smooth the transition between security labels, we can apply a trapezoidal function

$$\text{trapmf}(x; a, b, c, d) = \max(\min(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}), 0)$$

and define four different membership functions for each security label (See Figure 1).

- unclassified:$uc(x) = \text{trapmf}(x; 500, 500, 550, 650)$
- classified: $c(x) = \text{trapmf}(x; 550, 650, 700, 800)$
- secret: $s(x) = \text{trapmf}(x; 700, 800, 850, 950)$
- top secret: $ts(x) = \text{trapmf}(x; 850, 950, 1000, 1000)$

Given a document, its membership degree to a specific security label is determined by its membership function. For instance, the membership degrees of a document with score are 600 are: 0.5 (unclassified), 0.5(classified), 0 (secret), and 0(top secret). Similarly,

[2]Even though there is a term "fuzzy" in Fuzzy MLS, the system does not adopt any concept from fuzzy inference.

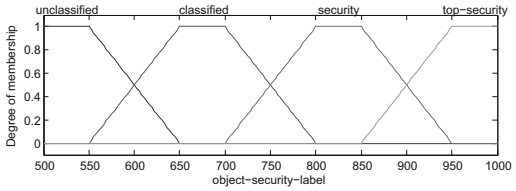

Figure 1: Object Membership functions

the membership degree of a document with score 601 are: 0.49 (unclassified), 0.51 (classified). We thus realize a smoother transition between security labels.

Assuming that we have a similar subject clearance score system and four different membership functions for subject security labels (see Figure 2).

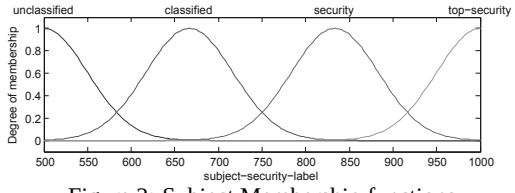

Figure 2: Subject Membership functions

Suppose also that we have a percentage risk score system and five membership functions for risk estimations, extremely low, low, medium, high, and extremely high, determined by experts (See Figure 3).

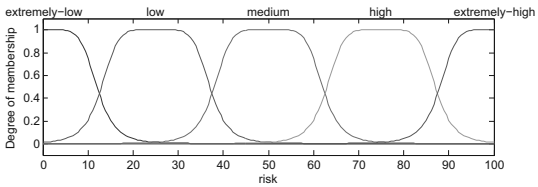

Figure 3: Risk Membership functions

We introduce some "*if* antecedent *then* consequent" rules in Table 1 to implement a risk-based BLP system to satisfy its simple security property. These rules determine an access risk mainly by the object security label and secondly by the subject security label. The interpretation of the rule with id equal to 1, for instance, is that *if* the object security label is unclassified, *then* the access risk is extremely low. The interpretation of rule with id equal to 2 is that *if* the subject security label is not unclassified *and* the object security label is classified, *then* the access risk is low. The last column in Table 1 represents the weight of a rule.

The procedures to evaluate the risk for a subject with score 750 that accesses a document with score 750 is as follows:

1. Fuzzification: this step calculates the membership degrees of the subject and the object for each different label based on their predefined membership functions. Subject membership degrees are: 0.0076 (unclassified), 0.5814 (classified), 0.5814 (secret), and 0.0076 (top secret). Document membership degrees are: 0 (unclassified), 0.5 (classified), 0.5 (secret), and 0 (top secret).

2. Application of fuzzy operations: this step calculates the firing degree of a rule based on the membership degrees and logical operations in the antecedent of the rule. In this particular example, we choose the *product* operation as the conjunction operation $T_p(x, y) = x \cdot y$. For instance, the confidence degree of the rule with id equal to 5 is $0.5814 \times 0.5 = 0.2907$.

Table 1: BLP risk inference rules

	Antecedent		Consequent	
ID	Subject Label	Object Label	Risk	W
1	N/A	unclassified	extremely low	1.0
2	not unclassified	classified	low	1.0
3	unclassified	classified	medium	1.0
4	unclassified	secret	high	1.0
5	classified	secret	high	1.0
6	secret	secret	low	1.0
7	top secret	secret	low	1.0
8	not top secret	top secret	extremely high	1.0
9	top secret	top secret	medium	1.0

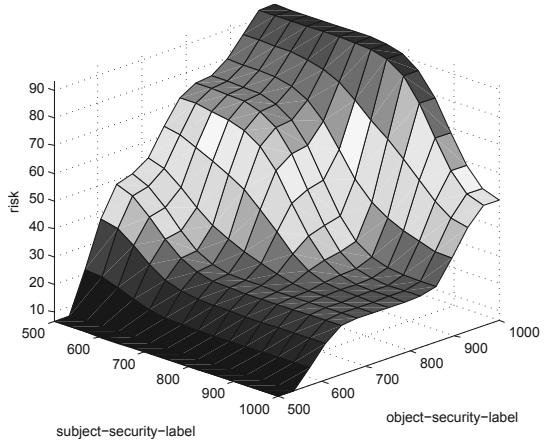

Figure 4: The surface of the risk-based BLP system

3. Application of the implication method: this step calculates the risk estimation of a rule based on the conjunction of its consequent, its firing degree, and its weight. For instance, the risk estimation of the rule with id equal to 5 is $0.2907 \times \text{mf}_{high} \times 1$, where mf_{high} is the membership function for "access risk is high".

4. Aggregation of all outputs: this step calculates the risk estimation based on the disjunction of the risk estimations of all rules. In this particular example, we choose the "sum" operation S_l as the disjunction operation $S_l(x, y) = \min(1, x + y)$. The result is a piecewise membership function, referred as the result function of aggregation *rf*.

5. Defuzzification: this step generates the final risk score by calculating the center of gravity of function *rf*.

$$\text{risk} = \text{centroid}(\text{rf}(x)) = \frac{\int \text{rf}(x) x \mathrm{d}x}{\int \text{rf}(x) \mathrm{d}x}$$

For this access, the final risk estimation is 38.6412.

Based on the choices of operations in the example, we obtain the surface of the risk estimation function shown in Figure 4. We can see that it follows the rules in Table 1.

As mentioned in the introduction, we have different choices of fuzzy operations. If we choose the conjunction, disjunction, implication method, and aggregation to be $\min(x, y)$, $\max(x, y)$, $\min(x, y)$, and $\max(x, y)$, respectively, the risk surface function, as shown in Figure 5, is different especially in some boundaries. However, the risk surface in Figure 5 seems to follow the rules in

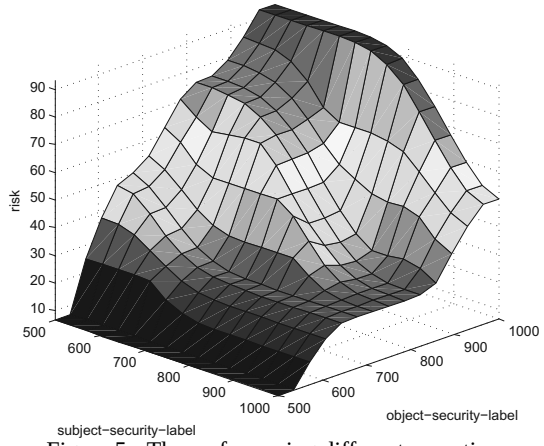

Figure 5: The surface using different operations

Table 1 as well. As a more precise comparison, given an access request to a document with score 750 from a subject with score 750, the inference system in Figure 4 estimates the risk at 38.6412, while the inference system in Figure 5 estimates the risk at 50. Such a non-trivial difference among risk scores results in different mitigation measures. We may want to know which score is more accurate. The answer is 38.6412 and the reason is detailed in Section 3.1.

2.2 Comparison

Compared to Fuzzy MLS, our Fuzzy BLP example suggests a more general and flexible approach to risk-based access control. The crucial difference is that we provide a general methodology to enable security officers to implement customized risk-based access control by specifying their own fuzzy rules. These rules can be translated from their experience in best practices. For Fuzzy MLS, the sigmoid function is specific to a MLS system. By specifying fuzzy rules like "*if* the difference between security labels is extremely high, *then* the access risk is extremely high", we can easily generate a risk estimation function by a fuzzy inference system that is similar to the sigmoid function applied in the Fuzzy MLS. If we wish to implement the same system as the Fuzzy MLS, we may train a Sugeno-type fuzzy inference system by the data generated by the Fuzzy MLS function [21]. The trained fuzzy inference system behaves exactly the same as the Fuzzy MLS. By contrast, without a fuzzy inference system, it is hard to construct the risk estimation function shown in Figure 4.

3. OPERATIONS ON RISK FACTORS

In the antecedent of risk rules, three operations are required: conjunction, disjunction, and negation. These operations are defined in a real unit domain [0, 1] and are used to aggregate membership degrees of different risk factors to generate the firing degree of the antecedent. We may consider the membership degree of a risk factor (a fuzzy set) to be a confidence degree as a member of the risk factor, and the firing degree of a rule to be a confidence degree of its antecedent. We thus use the term "confidence degree" to mean both "membership degree" and "firing degree".

These operations are usually sufficient to describe rules or recommendations from best practices. A conjunction relation between two risk factors means that these factors in a rule should be considered all together. A disjunction relation means that considering either of these two risk factors is fine for a rule.

A negation ($\neg$) means that the complement of a risk factor is applied in a rule, and thus is represented by $\neg x = 1 - x$ where x is the confidence degree of the risk factor. Unlike conjunction and disjunction, $1 - x$ is the only negation method widely used in practice and thus will not be discussed again.

Given an access request, several rules might be applicable, i.e. the confidence degrees of antecedents in these rules are not 0. Likewise, several different implication methods may be applied to generate the risk estimation for each rule and several different aggregation operations may be applied to generate the final risk estimation function.

In the following sections, we investigate the algebraic properties of these operations and illustrate the connection between their algebraic properties and the correlation between risk factors.

3.1 Conjunction

Intuitively, we expect that the conjunction (T) of two confidence degrees x and y of risk factors satisfies the following properties:

- The evaluation order does not matter when aggregating two or more confidence degrees.
- The aggregation $T(x, y)$ is non-decreasing with respect to both x and y.
- The value 1 (full confidence) is the identity element ($T(1, x) = x$), and 0 (no confidence) is the zero element ($T(x, 0) = 0$). The intuition is as follows. When combining 1 and another confidence degree x, because 1 means full confidence, the combined confidence degree should simply be x. When combining 0 and x, because 0 means no confidence, the combined confidence should be 0.

These requirements are met exactly by the following definition of t-norms.

DEFINITION 1 (TRIANGULAR NORM (T-NORM) [12]). *A binary operation T in the real unit interval [0,1] is a t-norm iff*

1. *it is associative and commutative, i.e. $\forall x, y, z \in [0, 1]$, $T(T(x, y), z) = T(x, T(y, z)$ and $T(x, y) = T(y, z)$;*
2. *it is monotonic in the first argument, i.e. $\forall x, y, z \in [0, 1], x \leq y$ implies $T(x, z) \leq T(y, z)$;*
3. *it satisfies the boundary condition, i.e. $\forall x \in [0, 1], T(1, x) = x$.*

LEMMA 1. *A t-norm T is monotonic in the second argument, i.e. $\forall x, y, z \in [0, 1], x \leq y$ implies $T(z, x) \leq T(z, y)$. Furthermore, a t-norm T satisfies $T(0, x) = 0$ and $T(x, y) \leq \min(x, y)$.*

All proofs of lemmas and theorems are in the Appendix. This lemma shows that the definition of t-norm satisfies all requirements for conjunction listed above.

There are uncountably many t-norms [16]. Different t-norms are desirable in different settings of risk factors. The following three basic t-norms are of particular interest to us, because they match some common correlations between risk factors.

- $T_g(x, y) = \min(x, y)$ (Gödel t-norm)
- $T_p(x, y) = x \cdot y$ (Product t-norm)
- $T_l(x, y) = \max(0, x + y - 1)$ (Łukasiewicz t-norm)

The Gödel t-norm is useful when x ad y are the confidence estimates of the same risk factor, perhaps obtained by two different methods of estimation or by two different experts/expert groups. Taking a pessimistic view, we take the minimal of the two estimates. For example, given a rule "*if* the object security label evaluated by DoD is extremely low *and* the object security label evaluated by CIA is extremely low, *then* the access risk is extremely

low", it makes sense to choose the minimal confidence degree from all estimates of the object in this rule. The Gödel t-norm is also the only t-norm where each $x \in [0, 1]$ is an idempotent element, that is, $T_g(x, x) = x$. This reflects the intuition that when two estimates are the same, one takes the consensus.

The product t-norm is useful when two confidence degrees from two risk factors are relatively independent from each other. For instance, given a rule "*if* the subject security label provided by DoD is not unclassified *and* the object security label by CIA is classified, *then* the access risk is low." Since the estimates of these two risk factors are based on sources that are highly independent from each other, a product t-norm should be adopted, like a similar situation in probability theory. *This is the reason why T_p is the best choice for the example in Section 2.1.*

The product t-norm belongs to an important subclass of t-norms called strict t-norms. A *strict t-norm* is strictly increasing in both of its arguments, that is, $T_p(x_1, y_1) > T_p(x_2, y_2)$ if either $x_1 > x_2 \wedge y_1 \geq y_2$ or $x_1 \geq x_2 \wedge y_1 > y_2$. Product t-norm, thus, generates smoother aggregation results than the other two t-norms do. If we have little information about the correlation between risk factors but want to obtain smoother access decisions, T_p is the best choice.

The Łukasiewicz t-norm is the only t-norm that satisfies the "Law of Non-Contradiction": $T_l(\neg x, x) = 0$. Such a characteristic is more useful when we need to specify a disjunction relation and thus will be revisited in Section 3.2.

LEMMA 2. *The Łukasiewicz t-norm is the pointwise smallest t-norm in these three t-norms.*

This characteristic is useful when two risk factors contribute to the risk assessment for an object. For instance, an improper access to a plasma-based weapon plan may represent two different values to an adversary: the value of the existence of such a plan and the value of its content, i.e. we lose money in both these respects. The risk of an access to the plan, thus, depends not only on each of these values but also on the sum of these values.

Accordingly, we may define a rule to control the access to these kinds of plans as "*if* a plan's existence value is low and the plan's content value is low, *then* the access risk is low" [3]. Because of the two different values represented by such plans, we will expect the t-norm to be able to better handle the case that neither of the two confidence degrees is really low but their conjunction should be very low. For instance, given some inputs, the confidence degree of one value to be low is medium, e.g. 0.5, and the confidence degree of another value to be low is medium too, e.g. 0.5. In this case, the confidence degree of the antecedent of the rule should be extremely low because the possibility of the sum of these two values *not* to be low is extremely high, i.e. this rule should be not applicable. In this case, T_p, T_g, and T_l will generate different confidence degrees: 0.25, 0.5, and 0, respectively. It is obvious that only T_l generates a reasonable confidence degree in this setting. The result is not surprising because T_l is point-wisely the smallest t-norms in these three operations. In other words, to reach a same confidence degree of a low risk, T_l requires higher confidence degrees of these two risk factors.

The Łukasiewicz t-norm is an example of another important subclasses of t-norms, the nilpotent t-norms. A *nilpotent t-norm* ensures that $\forall x \in [0, 1), \exists n \in \mathbb{N}$ such that $\overbrace{T(x, T(x \ldots, T(x, x)) \ldots)}^{n} =$ 0. That is, it has the following property: no matter how small a confidence degree a risk factor is, when enough risk factors with the same or even smaller confidence degrees are ANDed, the rule is no longer applicable because the confidence degree of its antecedent is zero.

[3]This example is only used to illustrate the use of T_l, therefore we do not use one concept to represent the sum of these two values, which may be simpler.

3.2 Disjunction

Intuitively, we expect that the disjunction (S) of two risk values x and y satisfies the following properties (only the last one differs from the conjunction case):

- The evaluation order does not matter when aggregating two or more confidence degrees.
- The aggregation $S(x, y)$ is non-decreasing with respect to both x and y.
- The value 1 (absolutely permit) is the subsuming element ($S(1, x) = 1$), and 0 (absolutely deny) is the identity element ($S(x, 0) = x$).

These properties are satisfied by any t-conorm. The definition of t-conorm is different from that of t-norm only in the boundary condition: $\forall x \in [0, 1], S(0, x) = x$. The standard way to define a t-conorm is to use a t-norm: $S(x, y) = 1 - T((1-x), (1-y))$ [16, 11], and the S is referred to as the T's dual t-conorm. Thus, the three dual t-conorms are:

- $S_g(x, y) = \max(x, y)$ (Gödel t-conorm)
- $S_p(x, y) = x + y - x \cdot y$ (Product t-conorm)
- $S_l(x, y) = \min(1, x + y)$ (Łukasiewicz t-conorm)

The criteria for choosing an appropriate t-conorm are exactly the same as for t-norms. The Gödel t-conorm is useful when aggregating the confidence estimates of the same risk factor. For instance, given a rule "*if* the object security label evaluated by DoD is extremely high *or* the object security label evaluated by CIA is extremely high, *then* the access risk is extremely high", it makes sense to choose the maximal confidence degree from all estimates of the object in this rule.

Product t-conorms can be used to calculate the disjunction of confidence degrees of two independent risk factors. Because either of these two factors is fine, the sum of the confidence degrees of two factors is a reasonable choice. However the part of confidence degree on which two independent risk factors impact together is counted twice; therefore the product t-conorm of two confidence degrees, referred to as x and y, should be $x + y - x \cdot y$. For instance, given the rule "*if* either a object security label is unclassified *or* a subject security label is top secret, *then* the access risk is extremely low", it makes sense to choose the product t-conorm as the disjunction in this rule.

As we mentioned in Section 3.1, the Łukasiewicz t-norm is the only t-norm that satisfies the "Law of Non-Contradiction". Likewise, the Łukasiewicz t-conorm is the only t-conorm that satisfies the "Law of Excluded Middle", that is, $S_l(\neg x, x) = 1$. It is usually the case that vague concepts do not follow the Law of Non-Contradiction. For instance, a document with a 750 score is "sort of" classified (0.5) *and* "sort of" secret (0.5); therefore, the confidence degree of the document to be classified and not classified is not zero.

However, many vague concepts still follow the law of excluded middle. For instance, a confidence degree of a document with a 750 score to be classified *or* not classified (e.g. secret) should be 1. Neither the Gödel t-conorm or the Product t-conorm can generate this correct answer, $\max(0.5, 0.5) = 0.5$ and $0.5 + 0.5 - 0.5 \times 0.5 =$

0.75. Only the Łukasiewicz t-conorm can generate the correct answer $0.5 + 0.5 = 1$. Therefore, the disjunction of a rule "*if* a object security label is classified *or* the object security label is secret, *then* the access risk is medium" should be the Łukasiewicz t-conorm. *This is the reason why S_l is the best choice in Section 2.1.*

3.3 Implication Operation

Given a rule "*if* x is A *and* y is B, *then* z is C", it is obvious that different choices of implication operations, i.e. the operation to generate the risk estimation of this rule based on its firing degree and its consequent C, may generate different risk estimations. Two commonly used implication operations are T_g and T_p. There are some good reasons why these two operations are the most widely applied.

In an auto cruise controller, a smooth controller makes passengers more comfortable. Likewise, a smooth risk decision maker may improve the information requesters experience because access decisions from an absolute permit and an absolute deny are a smooth transition.

Smoothness is a generic notion, and there are several different understandings and formalizations of what smooth means. In mathematical terms, the smoothness of a function is typically formalized as the existence of first, second, or higher order derivatives. To compare the smoothness of different functions in our context, such a definition is obviously not sufficient.

What we expect to be a smoothest operation is indeed a function that minimizes the number of disturbance and/or the sum of changing rates in either the "accelerating part" or the "decelerating part" in the output. Assume the firing degree of a rule to be x and its consequent to be C, its implication operation can be specified by a function $T(x, C) : [0, 1] \rightarrow [0, C]$ where T is a t-norm. To evaluate the smoothness of a function we thus introduce the following two measures.

- the number of indifferentiable points in the function domain [0,1], referred to as α-measure;
- the integral of the square of the second order derivative of the function in the function domain [0,1], i.e. $\int_0^1 T''(x, C)^2 \mathrm{d}x$, referred to as β-measure.

The following definition specifies how to compare the smoothness of two functions.

DEFINITION 2. *Let $T_i(x, C)$ and $T_j(x, C)$ be two implication operations, $T_i(x, C)$ is smoother than $T_j(x, C)$ iff*

- *the α-measure of $T_i(x, C)$ is smaller, OR*
- *the α-measure of $T_i(x, C)$ equals that of $T_j(x, C)$ and the β-measure of $T_i(x, C)$ is smaller.*

THEOREM 1. *T_p is the smoothest t-norm among all continuous t-norms.*

Based on this theorem, if we expect to obtain smooth access decisions between different risk estimations, we should choose T_p as the implication operation.

In a highly sensitive context, like for example in the case of CIA or FBI information, if something bad happens, we may want to take mitigating actions as quickly as possible. One way to achieve this goal is using adaptive fuzzy rules that take mitigation measures as the input of some risk factors and enforcing these rules as quickly as possible. The choice of implication operations can make a difference here. Because all t-norms are monotonic, to compare the quickness of different functions when enforcing a rule, we only need to introduce a new measure, γ-measure, which counts the minimal distance d of an implication function $T(x, C)$ such that $d = x' - 0$ and $T(x', C) = C$, i.e. how quick an implication can generate (enforce) a full fire degree of its consequent. If the d_x of T_x is less than d_y of T_y, we say that T_x is quicker than T_y.

THEOREM 2. *T_g is the quickest t-norm among all continuous t-norms.*

Based on this theorem, to quickly enforce some rules, e.g. emergency rules, we should choose T_g as the implication operation.

3.4 Aggregation

To generate the final risk estimation, we need to aggregate risk estimations from fired rules. Since each "IF THEN" rule only specifies a risk estimation based on certain risk factors, it is unsurprising that t-conorms (disjunctions) are used to aggregate risk estimations. Likewise, the choice of aggregation operations depends on the correlation between different rules. Different t-conorms S_g, S_l, and S_p can be applied to different situations.

Essentially, there is no difference between the aggregation of different rules and the disjunction used in the antecedent of a rule. A rule with a disjunction operation in its antecedent can be specified by two different rule. The example in Section 3.2, "*if* either a object security label is unclassified *or* a subject security label is top secret, *then* the access risk is extremely low", can be specified using two rules: "*if* a object security label is unclassified, *then* the risk is extremely low" and "*if* a subject security label is top secret, *then* the risk is extremely low". Obviously, S_l should be applied here to aggregate their risk estimation.

Therefore, the criteria discussed in Section 3.2 can be applied here to choose the best aggregation operation. If the relation is not clear, we may choose the best operation based on the requirement on the tolerance to inaccuracy inputs.

3.5 Discussion

One common situation in the applications of fuzzy inference systems, e.g. fuzzy control, is the choice of the fuzzy operation, e.g.T_p, T_l, or T_g. Such choice is somewhat arbitrary and, once selected, the chosen operation is applied to all rules. Since our objective is the best accuracy of risk estimation, we do *not* suggest using one type of conjunction, disjunction, implication, and aggregation for all rules. Instead, we suggest the following steps to choose the most appropriate operations in a fuzzy inference-based access control system.

1. Choose the best operations based on the correlation of risk factors, e.g., if two risk factors are independent to each other, T_p and S_p should apply.
2. Choose the best operations that meet desired properties for risk factors without clear correlations, e.g., if we prepare to quickly enforce emergency rules, we may choose T_g as the implication operation.
3. Choose the operations that require the least computation, i.e. T_p and S_p, if we do not have any special requirement. In some proposals [17], T_g and S_g were considered to be the cheapest one. However, based on our experimental results in Section 5.2, T_p and S_p are the cheapest one in a modern CPU like Intel Core 2 Duo.

4. CONTROLLING THE DAMAGE

There is one issue that exists in traditional access control systems but becomes more serious in risk-based access control systems. The issue is how to limit the damage caused by malicious users to a controllable scope, i.e. within an upper bound. Examples of potentially malicious users include employees who are going to be fired and may sabotage valuable resources, employees who are bribed by competitors, accounts that are hacked by intruders.

As mentioned in the introduction, the goal of risk-based access control is to maximize the flow of information provided that the total risk is kept under a certain level. When information flow is maximized, it is important that some post-access mitigation, e.g. post-obligations or audits that only enable administrators to take an action after accesses, are adopted. Generally speaking, a risk-based access control system provides more access opportunities for users than a traditional access control system does under same or similar situations. However, such a benefit has some cost. There is a non-trivial time window between accesses and the execution of post-obligations. This means that malicious users may possibly destroy or stole much more sensitive information during such a time window in a risk-based access control system than in a traditional access control system. Such a situation can arise in all risk-based access control systems, including the fuzzy inference-based solution described in this paper. In this section we propose a general approach to solve the problem. We also discuss how to implement the approach in a fuzzy inference-based system.

4.1 The Access Quota

Risk-based access control systems, when they are in operation, are somewhat similar to credit card systems. Users who are evaluated to be less risky can access information (buy anything using their credit cards) first and then fulfill obligations required by the access in time (pay their credit card balance in time). Therefore, it is natural for us to introduce a solution that is similar to a credit card system.

The solution introduces a new concept: the access quota that defines a number of access tokens. An access token is similar to a cent in a credit limit. There are two different types of access quotas: the access quota for users and the access quota for obligations. The access quota for users is the number of access tokens predefined for each user, similar to the credit limit for each credit card user. The number of token for a user is determined by security experts based on the background information about the user. Similarly, the number of tokens for an obligation is defined by security experts based on the importance or sensitivity of the obligation. The access quota for an obligation is similar to the price of a merchandise.

We do *not* define the access quota for any particular object because we believe that the access quota for obligations works better for our purpose. The access risk for an object depends on many factors including subjects and context. The access quota for an object may be different for different subjects and for different context factors. Therefore, there is no single number that can best describe the exact quota for a particular object. By contrast, each obligation is comparatively much simpler; the importance of each obligation is thus easier for experts to evaluate. The set of obligations required by each access are determined purely by the risk estimation of an access request; therefore, the sum of the access quotas of the obligations probably is the best way to describe the the access quota of the object in the access request.

The high-level idea of our solution to control the overall damage in a risk-based access control system is as follows (see Figure 6):

- All current access tokens of subjects are recorded in a tracking table of access tokens.
- Given an access request, the number of available access tokens of the subject of the access is verified after a risk estimation stage. If the number of available access tokens is less than the sum of the access quotas of the obligations required by the risk estimation, the access will be denied.
- If an access request is allowed, the tracking table is updated as follows: the access tokens of the subject of the access are subtracted by the sum of the access quotas of the obligations required by the access.
- If an obligation has been fulfilled, the tracking table is updated as follows: the access tokens of the subject of the access that is relevant to the obligation are added by the access quota of the obligation.

The quotas for both users and obligations depend on applications and may vary. In some implementations, we may further set day, week, or month quotas for users or may assign quotas for obligations by some more complicated equations to realize the fine-grained control of quotas. Even though a real system based on our approach could be fairly complicated, if we assume that the maximum quota for users is α, the minimum quota for obligations is β, and the most valuable object costs γ, we have the following theorem to prove the effectiveness of our approach.

THEOREM 3. *If each risky access requires at least one obligation to be fulfilled, the potential damage resulting from any malicious user at any particular time has an upper bound* $\alpha\gamma/\beta$.

The assumption of this theorem is reasonable and fairly easy to be met in practice.

4.2 A Fuzzy Inference-based Solution

The solution introduced in the previous section obviously works for a fuzzy inference-based access control system as well. However, a more efficient solution can be devised based on the same idea in a fuzzy inference based system. The cost of such a solution is a slightly higher overall damage in the worst case.

The solution, referred to as Fuzzy Solution, is as follows:

- We introduce a new membership function $TokenCheck(s)$ where s is a subject such that

$$TokenCheck(s) = \begin{cases} 0 \text{ if } t(s) \leq \delta \\ 1 \text{ if } t(s) > \delta \end{cases}$$

  where $t(s)$ returns the current number of tokens of subject s and δ is a predefined threshold for a minimal number of required tokens. δ might be zero, the minimal quota of obligations, or the maximal quota of obligations.

- The body of each fuzzy inference rule is ANDed with the $TokenCheck(s)$ membership function. Such a step ensures that when the current number of tokens of the subject is below the threshold δ, all fuzzy inference rules become not applicable. A default deny will be returned in this situation.

If we adopt the same assumption of Theorem 3, then we have the following lemma about the Fuzzy Solution.

LEMMA 3. *If each risky access requires at least one obligation to be fulfilled, the potential damage resulting from any malicious user at any particular time based on a Fuzzy Solution has an upper bound* $(\alpha/\beta + 1)\gamma$.

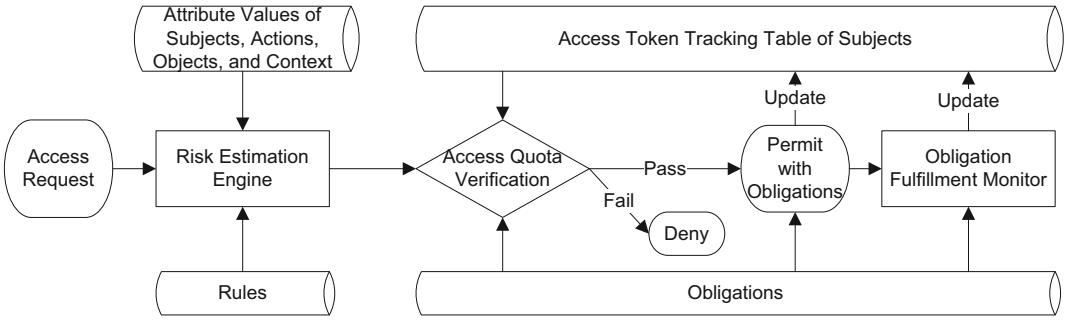

Figure 6: A General Method to Control the Overall Damage

Due to the expressiveness of membership functions and fuzzy rules, we may implement more flexible policies to control the overall damage. One case would be a better solution to control the overall damage resulting from high level managers who typically own much large access quotas. The solution is based on a limitation on the hourly token changing rate of a subject, i.e. no subject can spend too many access tokens within one hour. [4]

Such a solution only requires a new table to record the update history of access tokens, and a new membership function $TokenRate(s)$ to calculate an access token hourly changing rate based the table and compare the rate with a threshold θ. $TokenRate(s)$ returns 0 if the rate is larger than θ and returns 1 otherwise. The $TokenRate(s)$ will be ANDed with all fuzzy rules to enforce this policy.

5. THE EFFICIENCY OF INFERENCE

There are various successful applications of fuzzy inference in engineering fields, such as the attitude control system of space shuttles [7] or the automatic focus systems of digital cameras [18]. Tens of parameters and hundreds of inference rules are usually sufficient for these applications. For instance, in the attitude control system, only three parameters and tens of rules are applied [7]. Therefore, the computation overhead is very small.

Because of the high expressiveness of fuzzy rules and membership functions, we believe that tens of parameters and hundreds of inference rules are also sufficient for majority risk-based access control systems. However, there is one crucial difference here. An access control system may provide services for tens, hundreds, or thousands users simultaneously, but a fuzzy controller is typically used to control one object's action, e.g. an instrument or a vehicle. Therefore, the computation requirement of the fuzzy inference may be a problem for a fuzzy inference based access control system in practice. In this section, we investigate this problem and present our answer to this important question based on experiments.

5.1 The Experiment Settings

Our objective is to verify whether the fuzzy inference is scalable enough for applications in risk based access control systems. Therefore, the required experimental data, including membership functions (and their parameters) and fuzzy inference rules, and inputs, are generated randomly [5] because we do not need to consider the rationale of these membership functions and inference rules. We are only concerned with their computation overhead.

[4]Perhaps a different time range could be better for some sensitive contexts. The hourly token changing rate is only used for an illustration purpose.

[5]The data generation cannot be purely random, for instance, it is highly possible that a random membership function is not valid due to illegal parameters.

As we can see from Section 2.1, the efficiency of a fuzzy inference system depends on the following factors:

- The scale of the inputs (risk factors).
- The scale of the inference rules (the knowledge of risk estimation).
- The complexity of the membership functions (the fuzzification of concepts).
- The complexity of the defuzzification methods.
- The complexity of the inference functions, e.g. the fuzzy operations in the rule body, the implication, and the aggregation of consequences.
- The complexity of the integral process. This is a hidden factor for the complexity of the fuzzy inference. The more accurate the integral result is, the high computation cost it requires.

To obtain a clear picture of the efficiency of a fuzzy inference system and investigate the relation between the efficiency and aforementioned factors, we first choose a typical setting of a fuzzy inference system for access control and then adjust each factor individually to obtain the results. The typical setting, which we believe meets most of the access control requirements, is as follows:

- Number of risk factors: 200.
- Number of inference rules: 3000.
- The conjunction, disjunction, implication, and aggregation are T_p, S_p, T_p, and S_p, respectively.
- Fuzzification method: the Gaussian curve function
$$f(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}}$$
- Defuzzification method: the centroid function.

The fuzzy inference system has been implemented using C Language and runs on a PC with Core 2 Duo 3.2GHz and 3GB Memory.

5.2 Experimental Results

The first experiment verifies the scalability of a fuzzy inference system based on the typical setting. We randomly generate different sets of access requests. The set size is in the range [200, 25600], and the total response time for each set is shown in a log-log graph (see Figure 7). It is unsurprising to see that the response time is linear in the size of request sets. Given the facts that 1600 requests and

3200 requests require 14.37s and 29.5s, respectively, to compute, it is safe to say that the computation power of modern computers is sufficient to execute fuzzy inference-based access control systems for a typical setting. In the following experiments, the number of access requests will be set to 800 if not otherwise specified.

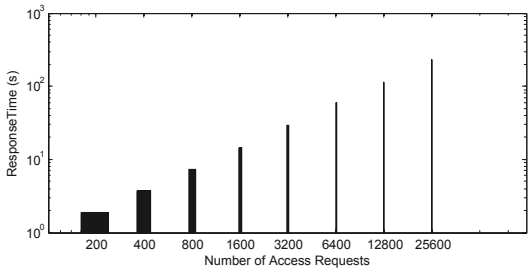

Figure 7: The scalability of fuzzy inference

The second experiment verifies the impact of the scale of risk factors and the result is shown in Figure 8. As we can see from Figure 8, the response time is unsurprisingly linear in the number of risk factors, which is good for our approach. More importantly, given 800 risk factors, a response time of 24s for 800 access requests is definitely sufficient for an access control system. Each access request only requires 0.03s to obtain a response.

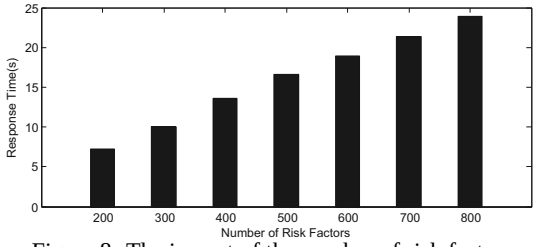

Figure 8: The impact of the number of risk factors

The third experiment verifies the impact of the scale of fuzzy rules, and the result is shown in Figure 9. The response time is linear in the number of fuzzy rules, which is reasonable. It is interesting to see that the impact of the risk factor scale is slightly more obviously than that of the fuzzy rule scale. Given 8000 fuzzy rules, a response time of 19s for 800 access requests is more than enough for an access control system.

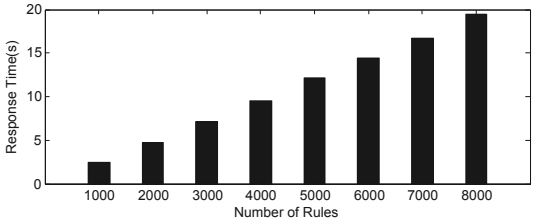

Figure 9: The impact of the number of fuzzy rules

The fourth experiment verifies the impact of different fuzzification methods (membership functions), and the result is shown in Figure 10. The membership functions in the figure are described as follows:

- dsigmf: a function composed of difference between two sigmoidal functions;
- gauss2mf: a Gaussian combination function;
- gaussmf: a Gaussian curve function;
- gbellmf: a generalized bell-shaped function;
- pimf: a π-shaped function;
- smf: a S-shaped function;
- trapmf: a trapezoidal-shaped function;
- trimf: a triangular-shaped function.

As expected, different fuzzification methods do result in quite different response times from 42.5s (gbellmf) to 4.65s (trimf). Fortunately, the most widely used Gaussian curve function (gaussmf) results in a reasonable response time of 7.21s. In practice, "cheap" functions may be applied in the situation in which a quick response is important or there are many concurrent access requests.

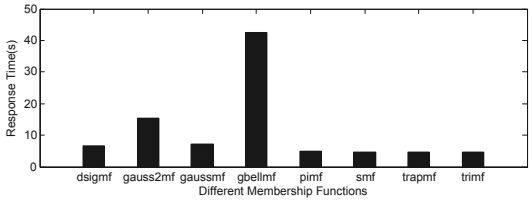

Figure 10: The impact of membership functions

The fifth experiment verifies the impact of different defuzzification methods, and the result is shown in Figure 11. The defuzzification functions in the figure are described as follows:

- centroid: the centroid of area;
- bisector: the bisector of area;
- mom: the mean value of maximum;
- som: the smallest (absolute) value of maximum;
- lom: the largest (absolute) value of maximum.

Because both centroid and bisector require the inference system to calculate the integral of the aggregated consequence function, one would expect that both centroid and bisection result in a longer response time. However, all defuzzification methods show a similar performance. The reason why their results are similar is explained in next experiment. Because of similar computational complexity requirements, we are free to choose a defuzzification method that best serve our problem. Usually it is the centroid function.

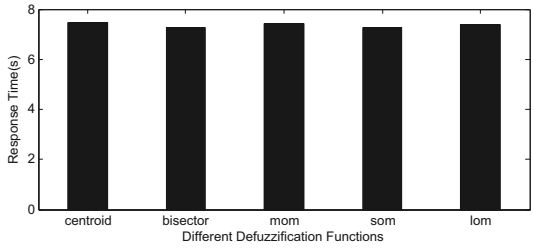

Figure 11: The impact of defuzzification methods

The implementation of defuzzification functions is based on sample points. There is no real integral process, and the integral is calculated by the sum of the sample interval multiplied by the function value at each sample point. Therefore, the computation complexities of different defuzzification functions are similar. The default size of sample points is 101. The sample size might greatly affect the response time. To investigate the impact of the sample size, we conduct an experiment. The result is shown in Figure 12. We can clearly see that the sample size has little effect on the response times because the cost of computing 10001 sample points is still

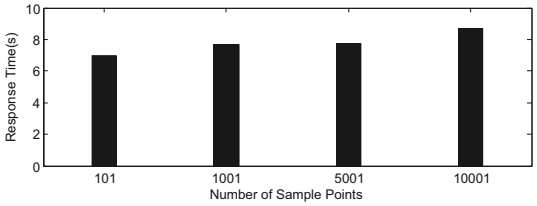

Figure 12: The impact of the size of sample points

dominated by the cost of other steps in the fuzzy inference. Such a result is good for our approach.

The last experiment focuses on the impact of different t-norms and their dual t-conorms in fuzzy inference. To see their difference, we decide that if a t-norm is chosen for the fuzzy conjunction, then the fuzzy disjunction, the implication, and the aggregation are its t-conorm, the t-norm, and its t-conorm, respectively. The experiment result is shown in Figure 13. In the literature of fuzzy inference, T_g and S_g were recognized as the cheapest fuzzy operations. The result shows that T_p and S_p are the cheapest one. The result also reveals a fact that a float computation is much cheaper than a comparison operation in a modern CPU.

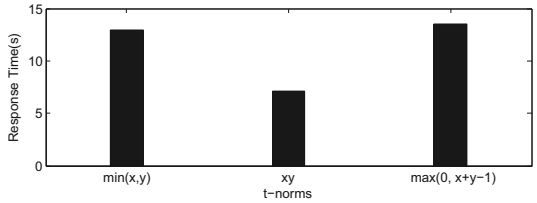

Figure 13: The impact of t-norms

6. RELATED WORK

We have already compared our work with the Fuzzy MLS system in Section 2.2. Therefore, in this section, we compare our work with relevant work in fuzzy logics and fuzzy control.

T-norms play a key role in all applications of fuzzy inferences; thus the construction and classification of t-norms is an important topic in fuzzy systems. Jenei et al. [15, 14] and Maes et al. [19] propose various different ways to construct t-norms based on their classification and algebraic properties. These proposals, though they are purely based some mathematical properties and have little connection with real applications, compliment our approach.

Properties of fuzzy operations have been investigated in fuzzy control in terms of stability [6] and in recommendation systems in terms of similarity [23]. Because their purpose is different from ours, these proposals adopt different measures to evaluate different operations. One property mentioned in this paper, that is, the smoothness, has been investigated in fuzzy control [5]. However, the distinct feature of smoothness in fuzzy control approaches is that these approaches have to reduce the oscillation before the output reaches its intended value when measuring these properties. Therefore, these approaches adopt different measures from ours, even if the meaning of smoothness is similar.

There are plenty of proposals on risk analysis, vulnerability assessment, threats and asset evaluation [3]. These proposals may help security officers to build membership functions and inference rules. The work reported in this paper complements these proposals by providing a sound approach that is able to translate these analysis results into executable actions and rules.

7. CONCLUSION

In this paper, we show that fuzzy inference is a good approach for estimating access risks. Specific problems concerning the application of fuzzy inference to access control are investigated and solved. In particular, the correctness of the solutions given in this paper are provable. In the future, we plan to apply our approach to database systems and investigate an approach to enable the database query engine to estimate query risks.

Acknowledgments

The work reported in this paper has been partially supported by the MURI award FA9550-08-1-0265 from the Air Force Office of Scientific Research.

8. REFERENCES

[1] FICO Credit Score, Apr 2009.

[2] C. Alberts and A. Dorofee. *Managing Information Security Risks: The OCTAVE (SM) Approach*. Addison-Wesley Professional, July 2002.

[3] C. J. Alberts and A. Dorofee. *Managing Information Security Risks: The Octave Approach*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

[4] J. Alcalá-Fdez, R. Alcalá, M. J. Gacto, and F. Herrera. Learning the membership function contexts for mining fuzzy association rules by using genetic algorithms. *Fuzzy Sets and Systems*, 160(7):905 – 921, 2009. Theme: Modeling and Learning.

[5] H. Allamehzadeh and J. Cheung. Smooth response sliding mode fuzzy control with intrinsic boundary layer. volume 1, pages 488–493 vol.1, May 2003.

[6] M. Benrejeb, A. Sakly, K. B. Othman, and P. Borne. Choice of conjunctive operator of tsk fuzzy systems and stability domain study. *Mathematics and Computers in Simulation*, 76(5-6):410 – 421, 2008. Mathematical Aspects of Modelling and Control.

[7] H. Berenji, R. Lea, Y. Jani, P. Khedkar, A. Malkani, and J. Hoblit. Space shuttle attitude control by reinforcement learning and fuzzy logic. In *Fuzzy Systems, 1993., Second IEEE International Conference on*, pages 1396–1401 vol.2, 1993.

[8] P.-C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In *IEEE Symposium on Security and Privacy*, pages 222–230. IEEE Computer Society, 2007.

[9] D. Dubois and R. R. Yager. Fuzzy set connectives as combinations of belief structures. *Inf. Sci.*, 66(3):245–276, 1992.

[10] M. J. Er and Y. Zhou. Automatic generation of fuzzy inference systems via unsupervised learning. *Neural Networks*, 21(10):1556 – 1566, 2008. ICONIP 2007.

[11] S. Gottwald. *A Treatise on Many-Valued Logics*, volume 9 of *Studies in Logic and Computation*. Research Studies Press Ltd., Baldock, Hertfordshire, England, 1st edition, 2001.

[12] P. Hájek. *Metamathematics of Fuzzy Logic*, volume 4 of *Trends in Logic*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1st edition, 1998.

[13] JASON Program Office. HORIZONTAL INTEGRATION: Broader Access Models for Realizing Information Dominance. Technical Report JSR-04-132, MITRE Corporation, McLean, Virginia 22102, 12 2004.

[14] S. Jenei. How to construct left-continuous triangular norms–state of the art. *Fuzzy Sets and Systems*, 143(1):27–45, 2004.

[15] S. Jenei. Recent advances in the field of left-continuous t-norms. In M. Stepnicka, V. Novák, and U. Bodenhofer, editors, *EUSFLAT Conf. (1)*, pages 23–24. Universitas Ostraviensis, 2007.

[16] E. P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*, volume 8 of *Trends in Logic - Studia Logica Library*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1st edition, 2000.

[17] V. Kreinovich, G. C. Mouzouris, and H. T. Nguyen. *Fuzzy Systems: Modeling and Control*, chapter Fuzzy rule based modeling as a universal approximation tool, pages 135–195. Kluwer, Boston, MA, 1998.

[18] C.-F. J. Kuo and C.-H. Chiu. Auto-focus control of a cmos image sensing module. *J. Intell. Fuzzy Syst.*, 18(4):405–415, 2007.

[19] K. C. Maes and B. De Baets. On the structure of left-continuous t-norms that have a continuous contour line. *Fuzzy Sets Syst.*, 158(8):843–860, 2007.

[20] D. H. Sharp and M. M. Wood-Schultz. QMU and Nuclear Weapons Certification What's under the hood? *Los Alamos Science*, (28):47–53, 2003.

[21] C.-T. Sun and J.-S. R. Jang. Using genetic algorithms in structuring a fuzzy rulebase. In S. Forrest, editor, *ICGA*, page 655. Morgan Kaufmann, 1993.

[22] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - i. *Inf. Sci.*, 8(3):199–249, 1975.

[23] A. Zenebe and A. F. Norcio. Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems. *Fuzzy Sets and Systems*, 160(1):76 – 94, 2009. Theme: Aggregation Operations.

[24] H.-J. Zimmermann, editor. *Practical Applications of Fuzzy Technologies*, volume 6 of *The Handbooks of Fuzzy Sets*. Springer, 2000.

APPENDIX

A. PROOFS

PROOF LEMMA 1. By the definition of t-norm, we have that $x \leq y$ implies $T(x,z) \leq T(y,z)$. Because t-norms is commutative, we have that $T(x,z) = T(z,x)$ and $T(y,z) = T(z,y)$. Therefore, $T(z,x) \leq T(z,y)$.

$x \leq 1$ implies $T(0,x) \leq T(0,1) = T(1,0) = 0$. Thus $T(0,x) = 0$.

$T(x,y) \leq T(1,y) = y$ and $T(x,y) \leq T(x,1) = x$. Thus $T(x,y) \leq \min(x,y)$. □

PROOF LEMMA 2. Based on Lemma 1, T_l is pointwisely smaller than T_g, i.e. $\min(x,y)$.

Since $x, y \in [0,1]$, we have that $(1-x)(1-y) \geq 0$, i.e. $x \cdot y \geq x+y-1$. Because $x \cdot y \geq 0$, T_l is pointwisely smaller than T_p. □

PROOF THEOREM 1. T_p is continuous, therefore the α-measure of T_p is zero.

$T_p'(x,C) = C$, and $T_p''(x,C) = 0$, therefore the β-measure of T_p is zero. □

PROOF THEOREM 2. It is obvious that $d = C$ for T_g because $T_g(C,C) = C$ and $T_g(x,C) < C$ for any $x < C$. For any other T, $T(C,C) \leq T_g(C,C) = C$. Based on Lemma 1 and T is monotonic, we have $T(x',C) = C$ iff $x' \geq C$. □

PROOF THEOREM 3. At any particular time, a user must have at most α tokens. The user in turn must have at most α/β suspending obligations, that is, obligations should be fulfilled but have not been fulfilled. Such a situation means that the user can only have at most α/β access requests given his access quota α. Therefore, the upper bound of the damages is $\alpha/\beta \times \gamma = \alpha\gamma/\beta$. □

PROOF LEMMA 3. The worst case in the Fuzzy Solution is that the threshold δ is zero. Such a situation means that even if the number of tokens of a user is not sufficient for an access request, it is still possible that the access is allowed. After the access, the number of tokens of the user become negative and the user cannot access any other resources before fulfilling his obligations. Therefore, the user can have at most $\alpha/\beta + 1$ accesses. Thus, the upper bound of his damages is $(\alpha/\beta + 1)\gamma$ □