

Reinforcing Access Control Using Fuzzy Relation Equations

Ali Berrached

André de Korvin

Department of Computer and Mathematical Sciences
University of Houston-Downtown
Houston, Texas 77002
berracheda@uhd.edu

Abstract

Current computer security systems are based on the premise that once a user presents valid credentials to the authentication system (e.g. valid ID and password), they are granted access permission to all resources assigned to the user that they claim to be. However, numerous studies have shown that most security breaches are done by unauthorized users impersonating as authorized users (e.g. by cracking or stealing passwords) or by circumventing the authentication system altogether (by exploiting security “holes” in the system). Once the authentication system is broken, the system and the information kept in it become wide open to unauthorized access and malicious usage. Moreover, because of the interdependencies among the various (computer and telecommunication) components of a distributed system, a security breach to one component can have repercussions throughout the system. The main objective of this paper is to present new security model that provides additional level of security checks based on heuristic information kept about various system components. The model allows a local host to evaluate and determine whether a remote request should be granted based on such information as the sensitivity level of the data being effected by the request, the type of request being made, and the probability of hostility of the user making the request. Typically, such information is very difficult to determine precisely since it depends on other attributes that are themselves imprecise or only partially known. The paper presents an algorithm for generating such fuzzy information based on their dependent attributes. The method is based on using basic rules of fuzzy set theory to establish a fuzzy relation between a set of dependent fuzzy quantities. The established relation can also be updated and adapted as the base information changes.

Keywords: Access-control, security, fuzzy-relations

1. Introduction

In recent years, with the widespread use of intranets and internets, users have become more and more

dependent on the services provides by networked systems where computer programs and potentially sensitive information are kept in (geographically) dispersed systems and exchanged over telecommunication facilities. Distributed systems have emerged to provide the means through which networked systems cooperate to process users tasks in a seamless and efficient fashion. Such systems provide tremendous benefits to their users but also raise new challenges. One of those challenges is to provide robust security mechanisms to guard against unauthorized access.

Security access control mechanisms play a key role in the overall structure of any security system. They are responsible for controlling the access permissions to system resources; i.e. determining who has access to which resource and with what type of access. Access control mechanisms rely on the authentication mechanisms to identify the users and ensuring that they are actually who they claim to be. The most common authentication method used to date is the user ID and password (or PIN number) combination, though other methods, such bio-metric identification, have been used with varying degrees of success [3].

The authentication system is clearly the cornerstone of current security systems. It also constitutes one of their main weaknesses. Indeed, current security systems are built on the premise that once a user presents valid credentials to the authentication system (e.g. valid ID and password), they are granted access permission to all resources assigned to the user that they claim to be. Numerous studies [3], however, have shown that a large number (if not most) of security breaches are done by unauthorized users impersonating as authorized users (by guessing passwords or stealing them through various means). Other security breaches occur by circumventing the authentication system altogether, by exploiting security “holes” in the system. Once the

authentication system is broken, the system and the information kept in it become wide open to unauthorized access and malicious usage. Moreover, because of the interdependencies among the various (computer and telecommunication) components of a distributed system, a security breach to one component can have repercussions throughout the system.

The objective of the proposed model is not to overhaul existing access control and authentication mechanisms but to provide an additional level of security checks. These security checks are triggered only after an access request is authenticated by the existing authentication system and the access is granted by the existing access control system. The proposed access control mechanisms differ from existing ones in that they are based on heuristic information about the user making the request, the sensitivity level of the resources that may be effected by the request, and the organization's tolerance to the type of losses that may be inflicted by granting the requested service. For example, a service request from a remote installation would be treated differently if the remote installation requesting the access is known not to provide a specific security service (e.g. secure authentication or firewalls). A security risk assessment would have to be performed by the local host (or its security guard) taking into account such factors as the remote host's security safeguards, the type of operations/services being requested, and the sensitivity of the information that may be effected by the remote access operations to determine whether the remote request should be serviced. The security risk analysis can be applied to any component of the distributed system (e.g. a user, an end-system, a communication link, a LAN, etc.) and would allow the local host to determine the level of security/hostility of the component.

The rest of the paper is organized as follows. Section 2 gives an overview of the proposed access control model and presents the risk assessment analysis used to determine whether a requested service should be safely granted. Section 3 presents an algorithm for estimating a fuzzy relation between a fuzzy set and its fuzzy attributes and applies it to the problem of computing the probability of hostility of a user in the context of the security access control model. Though the probability of hostility is used as an example, the algorithm is general and can be used in other applications as well.

2. Access Control Model

In this model we consider a user x_i from a defined set of users $X = \{x_1, x_2, \dots, x_m\}$ that have a desire to

perform a particular operation o_j on data set d_k , where o_j is an operation from the predefined set of operations $O = \{o_1, o_2, \dots, o_l\}$, and d_k is a data set from $D = \{d_1, d_2, \dots, d_m\}$ that is maintained by a given organization. In this framework, users are characterized by their probability of trustworthiness or hostility. The probability of hostility of a user can be determined based on several factors that can be deduced from information that is available about the user, such as the user's behavioral track record and the security services provided by the remote host. In addition, we define τ as the amount of damage that the organization can tolerate from servicing any user's request.

If Ph_i is the probability of user x_i being hostile, τ the amount of damage that an organization can tolerate, and EW_{jk} the worst loss expected from performing operation o_j on data d_k , then the expected loss E_{ijk} can be computed to determine whether user x_i should be allowed to perform operation o_j on data d_k . Obviously, the degree of the expected loss varies with the value of user hostility and is always less than or equals to the worst expected loss.

In general, information such as expected losses, user hostility, and allowable damage amount are very difficult to assess precisely in numerical terms. So it is natural to express them in the form of fuzzy sets. In linguistic terms, a user can be defined as very hostile, somewhat hostile, or not hostile, and the amount of damage can be expressed as very high, low, very low [4]. In fuzzy expressions, somewhat hostile, for instance, can be expressed as:

$$Ph = .9/.2 + .8/.1$$

where the supports (i.e. .2 and .1) are the probabilities of the user being hostile and the .9 and .8 are the membership values. Damage amount can also be expressed in a similar fashion, with the supports being expressed in dollar units (see [1] for detailed illustration).

We establish a procedure to determine whether a user x_i should be allowed to perform some operation o_j on some data d_k as follows:

1. Compute the expected loss E_{ijk} by evaluating $EW_{jk} \cdot 5Ph_i$, where the 5 operator is defined on any two fuzzy sets A and B as:
2. Apply Jain's method [2] of comparison to compare E_{ijk} to τ by constructing the *maximizing set* M of fuzzy sets E_{ijk} and τ . Given two fuzzy set A and B, their maximizing set is the fuzzy set that contains all the supports from A and B with the

degree for each support is the ratio of the support itself to the maximum support of A and B. For example, if $A = 0.4/10 + 0.7/30 + 0.2/40$ and $B = 0.1/15 + 0.8/50$ then the set of supports of A and B is $\{10, 15, 30, 40, 50\}$. Hence the maximum support is 50. Therefore, the maximizing set M of A and B is:

$$M = \frac{10}{50}/10 + \frac{15}{50}/15 + \frac{30}{50}/30 + \frac{40}{50}/40 + \frac{50}{50}/50$$

Hence:

$$M = 0.2/10 + 0.3/15 + 0.6/30 + 0.8/50 + 1/50$$

3. Compute $E_{ijk} \wedge M$ and $\tau \wedge M$ then make a comparison between the sets. If the greatest membership value of $E_{ijk} \wedge M$ is greater than the greatest membership value of $\tau \wedge M$, then the request is not granted since the expected loss from allowing user x_i to perform operation o_j on the data d_k is larger than the allowable total loss that he organization can tolerate. On the other hand, if the greatest membership value of $E_{ijk} \wedge M$ is less than the greatest membership value of $\tau \wedge M$, then the request is granted. The reader is referred to [1] for more detailed discussion of this algorithm.

3. Computing the Probability of Hostility

In this section, we present an algorithm for estimating the fuzzy probability of hostility Ph. It is clear that Ph is dependent on attributes of the user and the remote host from which the request is made. Such attributes may include such factors as (1) whether the user is making the request from a trusted site (e.g. whether the remote site is known to have robust security safeguards against password cracking, packet sniffing etc.), (2) whether the local host itself has robust safeguard (obviously, the more robust the local security is the less likely a hostile user is able to break-in), and (3) whether the request matches the user's behavioral history of usage (e.g. if the user is accustomed to access the system between the hours of 9am and 5pm and the request is made at 3am, that should raise the suspicion). Obviously, the actual list of attributes will differ from one system to another based on the specific needs and circumstances of each system.

3.1 Defining the Association

Let \mathbf{X} and \mathbf{Y} be two finite sets and R a fuzzy relation from \mathbf{X} into \mathbf{Y} . Note that R is a fuzzy subset of the Cartesian product $\mathbf{X} \times \mathbf{Y}$ and can be represented by a matrix with coefficients in the interval $[0,1]$, where

the general entry $R[x,y]$ is defined by the membership of (x,y) in R, for all $x \in \mathbf{X}$ and $y \in \mathbf{Y}$.

Let A and Ph be two fuzzy membership functions in \mathbf{X} and \mathbf{Y} , respectively. We define the \diamond operator, for all $y \in \mathbf{Y}$, as:

$$[A \diamond R](y) = \sup_{x \in \mathbf{X}} [A(x) \wedge R(x,y)]$$

In our case, \mathbf{Y} denotes a finite set of probabilities from which Ph draws its values. For example, $\mathbf{Y} = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. And \mathbf{X} denotes the set of attributes that effect Ph. For example, we can define \mathbf{X} to consist of the following attributes:

- Remote Host Trusted (TR)
- Remote Host Hostile (HR)
- User has good behavioral history (GH)
- User has bad behavioral history (BH)

Note that $A \diamond R$ is a fuzzy subset of \mathbf{Y} . Moreover, based on our definition of the \diamond operator, $[A \diamond R](y)$ represents the strongest support to the belief that Ph equals y. This stems from the sup-min composition and can be intuitively depicted in Figure 1 for the above example where $x_1, x_2,$ and x_3 denote some attributes from the set \mathbf{X} . Note in the figure that the strength of each link between y and A is equal to the strength of its weakest segment (min. operation). The strength of the association between y and A is

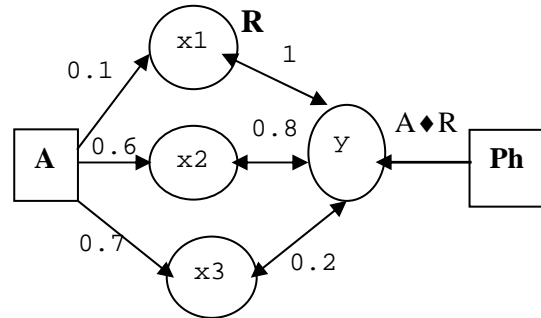


Figure 1: Graphical interpretation of the \diamond operator

equal to the strength of its strongest link (sup operation). Therefore, we conclude that, for all $y \in \mathbf{Y}$, $Ph(y) = [A \diamond R](y)$; or more concisely:

$$Ph = A \diamond R \quad (1)$$

Equation (1) allows us to compute the probability of hostility Ph for a user given a fuzzy relation R and a set of attributes A for that user. Conversely, given the probability of hostility Ph and a set of attributes A for some user x, one can solve equation (1) for R. We

note that equation (1) may have more than one solution; i.e. more than one relation may satisfy the equation for a given pair of A and Ph. However, the maximal fuzzy relation that satisfies equation (1) is given as [5]:

$$\hat{R} = A \varphi Ph \quad (2)$$

where φ is the fuzzy implication operator defined on $[0,1] \times [0,1]$ as:

$$a \varphi b = \sup\{c \mid 0 \leq c \leq 1, a \wedge c \leq b\}$$

Equation (2) provides the strongest association between A and Ph that satisfies equation (1). Note also that relation \hat{R} can be interpreted as a set of fuzzy inference rules between A and Ph.

3.2 Estimating the Maximal Fuzzy Relation

The previous section established an equation (equation (1)) for computing the probability of hostility Ph for a user with a set of attributes A, given a fuzzy relation R between A and Ph. Equation(2) gives the maximal fuzzy relation between a pair of A and Ph that satisfies equation (1). In order to find a fuzzy relation R that can be used to compute the probability of hostility for any user given his/her set of attributes, we need to establish a correspondence between certain sets of attributes and certain values of Ph. The estimation problem can be formulated as follows:

Given a set of pairs of fuzzy sets $(A_1, Ph_1), (A_2, Ph_2), \dots, (A_N, Ph_N)$, estimate R such that the system of equations:

$$A_k \blacklozenge R = Ph_k \quad (3)$$

is satisfied for all $k=1,2,\dots,N$

The given set of pairs can be viewed as a training set based upon which R is estimated. Assuming the training set is well designed, the estimated relation can be used to compute the probability of hostility Ph for any user, given the user's set of attributes A.

As discussed in the previous section, each individual equation in the system of equations (3) may have more than one solution with the maximal solution given by equation (2). Let \mathfrak{R}_k be the set of relations that satisfy the k^{th} equation of the system:

$$\mathfrak{R}_k = \{R \mid A_k \blacklozenge R = Ph_k\}$$

Note that if $\bigcap_{k=1}^N \mathfrak{R}_k = \phi$, then the system of equations has no solution. However, if

$$\bigcap_{k=1}^N \mathfrak{R}_k \neq \phi \quad (4)$$

it can be shown [5] that the maximal relation that satisfies the system of equations can be computed by intersecting all the fuzzy maximal relations that satisfy the individual equations; i.e.:

$$\hat{R} = \bigcap_{k=1}^N \hat{R}_k$$

where $\hat{R}_k = A_k \varphi Ph_k$.

From a practical point of view, there is no simple way to verify condition (4). We therefore take the following three-step approach to estimating relation R:

- 1) Solve each equation of system (3) individually, using equation (1) to find the maximal relation $\hat{R}_k = A_k \varphi Ph_k$
- 2) Compute $\hat{R} = \bigcap_{k=1}^N \hat{R}_k$
- 3) Verify that \hat{R} as computed in step 2 satisfies each equation in system (3). If it does not satisfy each equation then it is not a solution; otherwise, \hat{R} is the maximal relation that can be established between the user attributes and the probability of hostility.

3.3 Example

To illustrate the above algorithm, let's assume \mathbf{X} and \mathbf{Y} are defined as follows:

$$\mathbf{Y} = \{0, 0.2, 0.4, 0.6, 0.8, 1\} \text{ and} \\ \mathbf{X} = \{\text{GH, BH, TR, HR}\}$$

(see section 3.1 for a more detailed definition of these attributes). We will use a vector notation for the fuzzy sets A and Ph, with the supports implicitly in the order specified in the sets \mathbf{X} and \mathbf{Y} above. For example, $A = \{0.9/\text{GH}, 0.1/\text{BH}, 0.8/\text{TR}, 0.4/\text{HR}\}$ is written as $A = [0.9, 0.1, 0.8, 0.4]$. For the sake of brevity, we assume a small training set consisting of the following two pairs of fuzzy sets:

- $A_1 = [0.9, 0.1, 0.9, 0.2]$
- $Ph_1 = [0.9, 0.7, 0.3, 0.2, 0.1, 0.1]$
- $A_2 = [0.1, 0.9, 0.1, 0.9]$
- $Ph_2 = [0.1, 0.1, 0.4, 0.5, 0.9, 0.9]$

Using equation (1), we get:

$$\hat{R}_1 = A_1 \phi Ph_1$$

$$\hat{R}_1 = [0.9, 0.1, 0.9, 0.2] \phi \begin{bmatrix} 0.9 \\ 0.7 \\ 0.3 \\ 0.2 \\ 0.1 \\ 0.1 \end{bmatrix}$$

$$\hat{R}_1 = \begin{bmatrix} .9\phi.9 .9\phi.7 .9\phi.3 .9\phi.2 .9\phi.1 .9\phi.1 \\ .1\phi.9 .1\phi.7 .1\phi.3 .1\phi.2 .1\phi.1 .1\phi.1 \\ .9\phi.9 .9\phi.7 .9\phi.3 .9\phi.2 .9\phi.1 .9\phi.1 \\ .2\phi.9 .2\phi.7 .2\phi.3 .2\phi.2 .2\phi.1 .2\phi.1 \end{bmatrix}$$

$$\hat{R}_1 = \begin{bmatrix} 1.7 .3 .2 .1 .1 \\ 1 1 1 1 1 \\ 1.7 .3 .2 .1 .1 \\ 1 1 1 1 .1 .1 \end{bmatrix}$$

Using the same procedure we find:

$$\hat{R}_2 = A_2 \phi Ph_2 = \begin{bmatrix} 1 1 1 1 1 1 \\ .1 .1 .4 .5 .9 .9 \\ 1 1 1 1 1 1 \\ .1 .1 .4 .5 .9 .9 \end{bmatrix}$$

Intersecting the two relations we get:

$$\hat{R} = \hat{R}_1 \cap \hat{R}_2 = \begin{bmatrix} 1 .7 .3 .2 .1 .1 \\ .1 .1 .4 .5 .9 .9 \\ 1 .7 .3 .2 .1 .1 \\ .1 .1 .4 .5 .1 .1 \end{bmatrix}$$

Finally, we verify whether \hat{R} is actually a solution; i.e. whether the following equalities are satisfied:

$$A_1 \diamond \hat{R} = Ph_1 \text{ and } A_2 \diamond \hat{R} = Ph_2$$

For example,

$$A_1 \diamond \hat{R} = \text{Sup} \left([.9, .1, .9, .2] \wedge \begin{bmatrix} 1 .7 .3 .2 .1 .1 \\ .1 .1 .4 .5 .9 .9 \\ 1 .7 .3 .2 .1 .1 \\ .1 .1 .4 .5 .1 .1 \end{bmatrix} \right)$$

Therefore, $A_1 \diamond \hat{R}$ is equal to:

$$\begin{aligned} & \begin{bmatrix} \text{Sup}[(.9 \wedge 1), (.1 \wedge .1), (.9 \wedge 1), (.2 \wedge .1)] \\ \text{Sup}[(.9 \wedge .7), (.1 \wedge .1), (.9 \wedge .7), (.2 \wedge .1)] \\ \text{Sup}[(.9 \wedge .3), (.1 \wedge .4), (.9 \wedge .3), (.2 \wedge .4)] \\ \text{Sup}[(.9 \wedge .2), (.1 \wedge .5), (.9 \wedge .2), (.2 \wedge .5)] \\ \text{Sup}[(.9 \wedge .1), (.1 \wedge .9), (.9 \wedge .1), (.2 \wedge .1)] \\ \text{Sup}[(.9 \wedge .1), (.1 \wedge .9), (.9 \wedge .1), (.2 \wedge .1)] \end{bmatrix} \\ &= \begin{bmatrix} .9 \\ .7 \\ .3 \\ .2 \\ .1 \\ .1 \end{bmatrix} = Ph_1 \end{aligned}$$

Having found a maximal relation R that satisfies the training set, we now can compute the probability of hostility for any user, given his/her attribute values.

3.4 Approximating the Maximal Fuzzy Relation

Based on our experimentation with the algorithm presented above, we noticed that in numerous cases the maximal relation \hat{R} obtained in step 2 of the algorithm does not satisfy the system of equations but produces values for Ph that are very close to the original values in the training set. In some cases, an approximate solution may be acceptable provided that the values of Ph generated by the obtained relation are within an acceptable margin from the original values provided by the training set. Let ϵ denote the (acceptable) margin of error for the membership values defined by Ph. An approximate solution to the system of equations for a given training set can be obtained by extending step 3 of the original algorithm as follows:

1) Solve each equation of system (3) individually, using equation (1) to find the maximal relation $\hat{R}_k = A_k \phi Ph_k$

2) Compute $\hat{R} = \bigcap_{k=1}^N \hat{R}_k$

3) If ($distance(Ph_k, [A_k \diamond \hat{R}]) \leq \epsilon$) for all $k = 1, 2, \dots, N$, then accept \hat{R} as a solution.

4. Conclusion

In this paper we presented an algorithm for reinforcing access control based on heuristic information about the user, the data being accessed, and the various system components. The model uses fuzzy set theory to assess the risks involved in granting the requested service based on uncertain/partial information. The algorithm establishes a relation between a fuzzy set and its dependent attributes. The algorithm can be used as a training algorithm to compute fuzzy sets based on a training set of the dependent attributes.

5. References

- [1] Berrached A., Beheshti M., De Korvin A., Hu C., and Sirisaengtaksin O., (1998), Computer Security Model Based on Uncertain/Partial Information, IPMU 1998, Paris, France.
- [2] Jain, R., (1977), A Procedure for Multiple Aspect Decision Making Using Fuzzy Sets, Int. J. System Science
- [3] Krause M. and Tipton F.H., (1998), Handbook of Information Security Management, CRC Press LLC, Boca Raton, Florida.
- [4] Palmer, I. C. and Potter, G. A., (1990), Computer Risk Management, Van Nostrand Reinhold, New York.
- [5] Pedrycz W. and Gomide F. (1998), An Introduction to Fuzzy Sets Analysis and Design, MIT Press, Cambridge, Massachusetts.