An Ad Hoc Trust Inference Model for Flexible and Controlled Information Sharing

Danfeng (Daphne) Yao
Computer Science Department
Rutgers University, New Brunswick
Piscataway, NJ 08854
danfeng@cs.rutgers.edu

***Abstract*—Pervasive computing allows data to be collected using sensors and mobile devices. Recent studies, however, show that in emergency and crisis situations conventional access control mechanisms are too rigid for information sharing. There is an increasing need to secure the information collected from the pervasive computing environments, and yet to be able to allow flexible data sharing to facilitate problem-solving and decision-making. Our work investigates the two seemingly contradictory factors, secure access and flexible adaptation, and designs a trust inference model for emergency and crisis situations. We describe an *ad hoc* trust inference model where access decisions are adaptive to the identity, history, and environment of a requester, for example, the degree of urgency. Our trust inference model is built on fuzzy logic. Our sharing control mechanism can also be applied to protect personal data and used for digital identity protection.**

Keywords: *ad hoc* trust inference, secure information sharing.

I. Introduction

Pervasive computing allows more data to be collected using sensors and mobile devices. To effectively utilize the vast amount of data generated, information needs to be shared across organizational and administrative boundaries. The purpose of authorization is to control and facilitate the access to shared resources by entities (people or devices) belonging to different autonomous domains. The challenge for security research on distributed authorization is two-fold: (1) secure and accountable access: how to guard the integrity and confidentiality of shared resources; (2) flexible adaptation: how to facilitate flexible and dynamic information sharing.

Conventional access control research has extensively addressed the need of secure access, focusing on comprehensive policy designs and analysis and efficient management of users' privileges and privacy. Research in crisis management [4], however, shows that in crisis situations (e.g., natural and technological disasters, terrorism, firefighting) traditional central command and control models are either unavailable or too rigid for urgent information sharing, and often fail to provide adequate supports for data access across organizational boundaries. There is an increasing need to secure the information collected from the pervasive devices (e.g., location information), and yet to be able to allow flexible sharing to facilitate problem-solving and decision-making. Cross-domain information sharing also requires high accountability, so that misuses of data can be discovered and malicious users can be identified and held accountable for their behaviors. These problems are unique and challenging in emergency and crisis situations because of the dynamic nature of shared data and users. One motivation scenario for our study is the ineffective inter-organizational crisis communication reality during Hurricane Katrina.

In this paper, we propose an *ad hoc* trust inference framework for crisis communication that supports flexible and secure information sharing across different administrative domains. Our goal is to support the automatic prediction of a requester's trustworthiness based on what is learned about the requester, including affiliation, identification, history, and context. The resource owner then determines the corresponding access privileges for the requester.

Studies from political science community have found that technologies can sometimes cause communication barriers during crises [4]. One of the reasons for this phenomenon is that protecting the integrity of information has been the main design goal in authorization systems. Conventional authorization systems are also designed to largely suit the need of intra-domain information access, e.g., requesters are typically employees of the organization. However, in crisis situations, the need for inter-organizational information sharing sharply increases, and access requests for sensitive data may come from outside the organization and from people who are not previously known. To meet the cross-domain information sharing requirements, the *status quo* is that one or multiple administrators are usually needed to be involved to give specific permissions to the outside users. For example, to allow a FEMA official to access the real-time location information of team members belonging to U.S. Coast Guard, the typical route is that one or more higher-level FEMA directors contact directly or indirectly via authorization letters administrators at U.S. Coast Guard to establish the collaboration and information sharing. This time-consuming manual process certainly cannot meet the fast-response need for information sharing required in crisis situations.

Ideally, in crisis situations, exceptions may be made to normal access rules may according to the specific conditions and scenarios. This step involves a logic process to evaluate the

tradeoffs of associated risks and benefits and is conventionally performed by a human administrator. For example, a U.S. Coast Guard official will assess the urgency of Hurricane Katrina and the benefits of sharing information with FEMA staffs, in order to decide whether or not to share location information to FEMA. There may not be access rules defined for this unique situation, therefore, logical human judgement is typically required according to the following patterns. If the FEMA staffs are trustworthy and the the rescue missions are urgent, then FEMA staffs are allowed to access the location information of U.S. Coast Guard members. Fuzzy logic system can be used to define and automate this logic process and therefore is particular useful for controlling information sharing in these open systems.

With the increasing use of GPS-enabled computing devices, location privacy has caught much attention in the security community. There are several different meanings for location privacy. One of them refers to how to enable a user to effectively and conveniently control the sharing of his location information. A challenging aspect of this problem is to enable sharing in dynamic collaboration environments, such as sharing among first-response teams who are not previously known to each other. The sharing and control of access need to be established dynamically in response to the need of crisis communications.

In this paper, we design a concrete fuzzy logic system for inferring trustworthiness for cross-domain information sharing in crisis situations. We identify and describe the key attributes involved in evaluating the trustworthiness of a requester, and define concrete membership functions for each fuzzy variable in the system. We illustrate the operations of aggregation and defuzzification for obtaining the final trust scores.

We design an audit mechanism for identifying cheating users (e.g., taking advantage of or manipulating context information) and fold the information into the trustworthiness computation to improve accountability. We propose to use a simple logging and auditing mechanism to monitor and adjust the accuracy of long-term trustworthiness predictions. The auditing process also provides incentives for users to behave well in the open systems and potentially deters lying.

II. Preliminary

In this section, we give the preliminary knowledge for our *ad hoc* trust inference system. We briefly introduce the key concepts in fuzzy logic.

Fuzzy logic, unlike the conventional crisp logic, is defined as the logic system that uses imprecise or uncertain inputs to infer outputs. The arts of fuzzy systems were first proposed by Lotfi A. Zadeh in 1965 [19]. They became widely used in commercial applications such as subway systems, electronic appliances, and trading systems, in late eighties and early nineties. Fuzzy systems collectively refer to fuzzy sets, logic, algorithm, and control. The fundamental idea behinds all fuzzy systems is a quite simple one: the transition from one output state (e.g., 0) to the other (e.g., 1) is gradual and continuous,

$$\text{Earliness}(x) = \begin{cases} 1, & \text{IF time}(x) \leq 1200, \\ \frac{2000-\text{time}(x)}{800}, & \text{IF } 1200 < \text{time}(x) \leq 2000, \\ 0, & \text{IF time}(x) > 2000 \end{cases}$$

Time of the day	Degree of earliness
0900	1
1400	0.75
1600	0.5
2200	0

Fig. 1. An example of membership function and degrees of membership.

which is contrary to abrupt and crisp changes between zero and one.

In general, fuzzy systems can be used for approximate reasoning, where the inputs and the parameters of a system are incomplete, inaccurate, or imprecise. Fuzzy logic makes estimated decisions with inputs that have degrees of fuzziness. Existing applications of fuzzy logic take advantages of its efficiency and are usually more efficient compared to nonfuzzy methods in terms of computational costs. To develop a fuzzy logic system, one needs to identify the inputs and outputs and their ranges, define membership functions for the variables, construct fuzzy rule sets, and fine-tune the systems.

We give a simple example to illustrate the use of fuzzy logic. Consider *the time of the day* as the only fuzzy variable, and the output is *the earliness*. Suppose the membership function is defined as in Figure 1. Given this definition, some example values of earliness are shown in Figure 1.

III. Our Ad Hoc Trust Inference Model

In this section, we design an *ad hoc* trust inference model that allows a resource owner to infer the trustworthiness of a requester in an *ad hoc* fashion. There are two main players in our *ad hoc* trust inference model, a *resource owner* and a *requester*. We assume that the requester may be malicious and submitting false information to the resource owner in order to gain access. We do not assume any prior trust relationships between the resource owner and the requester (thus the name *ad hoc*), i.e., they may not know each other. The key point in our *ad hoc* trust inference model is that the trustworthiness is computed based on the profile of a requester, rather than from a single attribute. The profile of a requester captures several facets of the user or his or her organization. The elements in the user's profile are integrated using fuzzy logic rules and are collectively evaluated to make access decisions. As a result, the final access decision does not depend solely on any single input. The less rigid structure of fuzzy inference rules allows flexible-yet-controlled access decisions, namely, a partial access decision that is between 0 and 1.

A. Overview and Setup of Ad Hoc Trust Inference

Before a resource owner performs *ad hoc* trust inference, it needs to go through a *setup phase*. In the setup phase,

the resource owner gives the definitions to several important components of the fuzzy logic system including attributes, fuzzy variables, membership functions, and fuzzy rule set. The details of how to define these fuzzy logic components are to be described one-by-one in the following sections.

1) Define attributes from which trustworthiness may be inferred.
2) Define the fuzzy variables associated with each attribute. See Table I.
3) For each fuzzy variable, define a membership function. See Section III-C.
4) Define the output membership function for the output variable (i.e., degrees of trustworthiness).
5) Define fuzzy rules to specify the logic used to infer the trustworthiness score from attributes.

Before we dive into that topic, let's give a brief overview on the procedure of *ad hoc* trust inference. Our *ad hoc* trust inference procedure is run by the resource owner and consists of five main steps: FUZZIFICATION, RULE_APPLICATION, AGGREGATION, DEFUZZIFICATION, and AUTHORIZATION. The inputs are n crisp values $(x_1, \ldots, x_n)$, where $x_i \in [0, 1](1 \leq i \leq n)$ is a numerical attribute value defined in Section III-B. For the output, a crisp numerical value $y \in [0, 1]$ is computed representing the inferred trustworthiness score.

1) FUZZIFICATION: For each input, compute the degrees of membership based on the membership functions.
2) RULE_APPLICATION: Apply fuzzy logic rules to the inputs and obtain a conclusion for each applicable rule.
3) AGGREGATION: The conclusions are combined into a logical sum.
4) DEFUZZIFICATION: A firing strength for each output membership function is computed. Combine these logical sums in a defuzzification process to produce a crisp trust score.
5) AUTHORIZATION: Based on the computed trust score and the sensitivity of requested information, the resource owner determines the access permission of the requester.

In the following sections, we describe our *ad hoc* trust inference system in details.

B. Attributes and Fuzzy Variables

Our inference model for computing trustworthiness is based on three types of attributes associated with a request as shown in Table I.

Definition 1: In our *ad hoc* trust inference model, an *attribute* describes a property of a request or the person who submits the request.

In our model, an attribute takes a numerical value (e.g., 0, 0.5, or 1) and is associated with several fuzzy variables, which are defined below. For example, an attribute *urgency level* may have value 0.9.

Definition 2: In our *ad hoc* trust inference model, a *fuzzy variable* is a linguistic value (i.e., a word or a phrase and usually an adjective) that describes and characterizes the numerical attribute value. An attribute may have multiple fuzzy variables. Our *ad hoc* trust inference system has five fuzzy variables for all attributes and for the output: very high, high, medium, low, and very low.

For example, our attribute *urgency level* has five fuzzy variables: very high, high, medium, low, and very low. As it will soon become clear, an attribute value (e.g., 0.9) will be mapped to several fuzzy variables (e.g., high, medium, low) according to membership functions. Alternatively, fewer fuzzy variables may be defined in a coarser granularity, e.g., high, medium, low.

We define three types of attributes in our model: *identity*, *history*, and *environment*. We also consider how to validate the authenticity of the attribute values that are submitted by the requester. Our categorization of attribute types groups properties of a request and captures the necessary information in order to determine the trustworthiness of a requester. Nevertheless, we do not claim that our list of attributes described next is comprehensive, as more attributes may be introduced according to specific applications. For example, *ranking* may be added to be an identity type attribute and *connection security* may be added to be an environment type attribute. For the clarity of presentation, we choose to omit them in this paper.

- *Identity type* includes attribute affiliation score. *Affiliation score* is an attribute representing the trustworthiness of an organization. Higher scores mean higher trustworthiness or trustworthy relationship. The score is determined based on the home organization (i.e., main affiliation) of a requester and the relationship standing of that organization with the resource owner. For example, U.S. Coast Guard gives FEMA members the affiliation score of 0.8 out of 1. Bob is a FEMA member and thus has the affiliation score of 0.8. A default score may be given if the requester's home organization is unknown to the resource owner. Audit results may be used to dynamically adjust affiliation scores assigned to organizations, and will be discussed in more details later. Identity attribute can be authenticated with digital credentials (e.g., role credentials) submitted by the requester.
- *History type* includes attribute previous performance. *Previous performance* contains the information about a requester or his organization that is derived from the history of interactions with the resource owner. Higher attribute values mean higher or better previous performance. There are several methods to evaluate previous performance. For example, a simple approach is to compute $\frac{\text{Number of good transactions}}{\text{Number of bad transactions}}$. Due to space limit, we do not delve into this topic in our paper [1].
  Previous transaction history is usually kept by the resource owner and is not submitted by the requester. Therefore, the attribute value is computed by the resource owner and there is usually no need to validate the attribute value. Reputation information is typically gathered from

[1]If there is no prior interaction between the resource owner and the requester, the resource owner may assign a default value to this attribute.

peers of the resource owner.

- *Environment type* includes the attribute urgency level. *Urgency level* is an attribute whose value is specified by the requester and defines how urgent a requester needs the information. Higher attribute values mean higher urgency. Because the urgency level is *self-claimed*, it may or may not reflect the real situation, e.g., a user may falsely claim his request is extremely urgent in order to receive higher trust score and authorization. To catch this type of cheating activities, our model requires an audit mechanism to monitor the truthfulness of self-claimed urgency levels and provides feedback to the trust inference process. For example, if a user or a group of users has been consistently exaggerating the urgency levels of requests, then this information will be incorporated into previous performance attribute and affiliation score attribute. Thus in future requests, a cheating user will be penalized.

The above categories of attributes are factors to be used to determine a requester's trustworthiness. The sensitivity level of the requested information is not included in these attributes as it is independent of a request. However, this sensitivity level should be used to determine a requester's access authorization.

We call the output of trust inference model as *trust score*. The output is also associated with multiple fuzzy variables (e.g., {very high, high, medium, low, very low} in our model).

As shown in Table I, attribute affiliation score can be a value between 0 and 1, and can be mapped to five fuzzy variables according to the membership functions of the fuzzy variables. The range is chosen arbitrarily in this paper.

Membership functions will be defined in order to fuzzify an attribute value to multiple fuzzy sets. A fuzzy rule set is also defined to infer a set of trustworthiness values of a requester from the fuzzified attribute values. The inferred trustworthiness values are then aggregated and defuzzified to obtain the final crisp score. More details of this process are described next.

C. Membership Functions

In fuzzy theory, a membership function defines to what degree a variable belongs to a fuzzy set. Formally, a fuzzy set is defined as follows. We call the process of mapping a fuzzy variable to its membership of a fuzzy set as *fuzzification*.

Definition 3: A *fuzzy set* is a pair (X, m) where X is a set and $m : X \rightarrow [0, 1]$. For each $x \in X$, $m(x)$ is the degree of membership of x.

If an element is not included in the fuzzy set, then $m(x) = 0$; if it is a fully included member, then $m(x) = 1$. Fuzzy members are characterized by values that are between 0 and 1.

In our trust inference model, a membership function is defined for each fuzzy variable. Our model has five fuzzy variables {very high, high, medium, low, very low} for our three types of attributes.

There are several commonly used membership functions. For the ease of illustration, we choose a triangular shape membership function with height of 1 as shown in Figure 2. Bell-shape membership functions are also widely used in many fuzzy logic systems that give nonlinear (e.g., quadratic) transitions between 0 to 1.

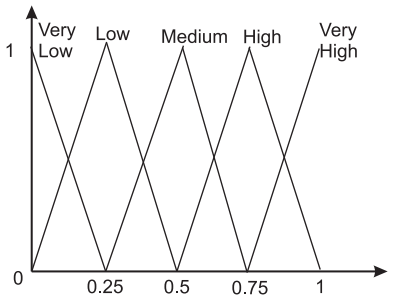

Fig. 2. Triangular shape membership functions of 5 fuzzy variables {very high, high, medium, low, very low}. The X-axis represents the crisp input. The Y-axis represents the degree of membership.

Once membership functions are defined for fuzzy variables, a crisp input can be fuzzified to obtain degrees of membership for all the fuzzy variables. Let's illustrate the process in the following example.

Example 1: U.S. Coast Guard (USCG) is the resource owner. Bob is a FEMA member who requests to access the location information of certain USCG personnel. USCG gives FEMA members the affiliation score of 0.8. Bob has never requested information from USCG before (i.e., no previous transaction history). U.S. Coast Guard assigns a default value 0.4 as the previous performance attribute. Bob claims that his request is urgent and he gives 1 to the urgency level attribute.

Using the membership functions in Figure 2, we obtain the degrees of membership for each attribute, which is shown in Table II.

Once the degrees of membership of each crisp input are computed, fuzzy rules are to be applied as presented next.

D. Fuzzy Rule Sets

Fuzzy rule sets are defined in the IF-AND-THEN form as follows.

$$R^i: \text{ IF } x_1 = A_1^i \text{ AND } x_2 = A_2^i \text{ AND } \ldots \text{ AND } x_n = A_n^i, \text{ THEN } y = B_i.$$

The rules and the number of rules to be defined may based on the specific applications and administrative policies of the resource owner. To illustrate how fuzzy rules can be defined for our attributes, we give several examples of *ad hoc* trust inference rules in a table format in Table III.

In our setup, each attribute including the output contains five fuzzy variables. In order to enumerate all the combinations of fuzzy variables, it requires $4^5 = 1024$ number of fuzzy rules. However, fuzzy logic systems do not require all possible rules to be explicitly defined. A very complex system may contain just a hundred rules [9]. For our model, since the number of attributes is small, we expect that the number of rules in an actual prototype authorization system is not large. We give more discussions on this topic in our future work in Section V.

TABLE I
INPUT ATTRIBUTES AND OUTPUT IN *ad hoc* TRUST INFERENCE MODEL FOR CRISIS COMMUNICATIONS. AUTHENTICATION METHODS REFER TO HOW TO VERIFY THE CORRECTNESS OF ATTRIBUTE VALUES.

Attribute type	Attribute names	Ranges	Fuzzy variables	Authentication method
Identity	Affiliation score	[0, 1]	Very high, high, medium, low, very low	Digital credentials
History	Historic performance	[0, 1]	Very high, high, medium, low, very low	–
Environment	Urgency level	[0, 1]	Very high, high, medium, low, very low	Audit mechanism
Output	Trust score	[0, 1]	Very high, high, medium, low, very low	–

TABLE II
DEGREES OF MEMBERSHIP IN EXAMPLE 1.

Attribute name	Value	Very low	Low	Medium	High	Very high
Affiliation score	0.8	0	0	0	0.8	0.1
Previous performance	0.4	0	0.3	0.6	0	0
Urgency level	1.0	0	0	0	0	1.0

TABLE III
FUZZY RULES.

Attribute name	Affiliation score	Previous performance	Urgency level	Output trust score
Rule R_1	very high	medium	very high	very high
Rule R_2	high	low	very high	medium
Rule R_3	medium	high	medium	high
Rule R_4	low	low	very high	low
Rule R_5	very low	medium	very high	low

Compared to conventional predicate based logic rules, fuzzy rules are simple to define and intuitive to understand as it follows the natural logic of human mind. Therefore it can reduce the management difficulty for large complex systems, which in turn reduce the number of mistakes made by administrators in specifying the policies. How to specify fuzzy rules is also easy to learn that is important for training the personnel in government and military organizations.

Plug in our example membership values in Table II. Affiliation score selects rule R_1 and R_2. Previous performance and urgency level both select all rules except R_3. Because for attribute affiliation score the membership degrees of low and very low are zeros, R_4 and R_5 are not applicable. Therefore, from rules R_1 and R_2 the fuzzy outputs are *very high* and *medium*. *Fuzzy output* refers to the output fuzzy variable corresponding to a rule that is fired or has non-zero result. Next, we describe the aggregation and defuzzification steps.

E. Aggregation and Defuzzification

The aggregation step is to determine the firing strength of each rule and to combine the logical products for each rule. There exist several aggregation methods and the choice may be up to the resource owner. We illustrate the root-sum-square method in our paper. Because only rules R_1 and R_2 give non-zero results, the output fuzzy variable can be very high or high, correspondingly. For fuzzy variable *very high*, the firing strength denoted by P_1 is computed as $(0.1^2 + 0.6^2 + 1.0^2)^{\frac{1}{2}}$ = 1.1. For fuzzy variable *medium*, the firing strength denoted by P_3 is computed as $(0.8^2 + 0.3^2 + 1.0^2)^{\frac{1}{2}}$ = 1.3.

For completeness, we give the general formula for computing the firing degree P_i of a fuzzy variable f_i using root-sum-square method in Equation 1, where P_i denotes the firing strength of f_i, n is the number of (input) fuzzy variables, k is the number of rules that yield f_i as the output response, and d_{ij} denotes the degree of membership of input variable x_i in rule R_j.

$$P_i = \sqrt{\sum_{j=1}^{k} \sum_{i=1}^{n} d_{ij}^2} \tag{1}$$

The defuzzification step is to compute a crisp output by combining inference results using a fuzzy centroid algorithm, as specified in Equation 2, where C_i denotes the center point of f_i's membership function, P_i denotes the firing strength of a fuzzy variable f_i.

$$\text{Output} = \frac{\sum_{i=1}^{n} C_i \times P_i}{\sum_{i=1}^{n} P_i} \tag{2}$$

We obtain the output $(0.875 \times 1.1 + 0.5 \times 1.3)/(1.1 + 1.3)$ = 0.6 as the crisp output. Thus the inferred final trust score is 0.6. We have finished the descriptions of main fuzzy inference steps. For AUTHORIZATION, which is to determine whether a requester can access certain information, the resource owner makes the decision based on (1) the inferred trust score of the requester, (2) the sensitivity of requested information, and (3) the resource owner's local policies. Authorization policies may be defined following the conventional access control policies and are omitted due to page limit.

F. Auditing Mechanism

How a user judges a transaction as a bad or good transaction is usually specific to applications. For example, in peer-to-peer file sharing applications, a correct download from a peer in a timely fashion can be counted as a good transaction. For access control and information sharing scenarios such as what we

study, judging a transaction is good or bad is based on whether a requester is truthful in submitting his or her attributes. We propose to use an auditing mechanism to selectively monitor the transactions and provide the feedback to the inference process.

Each administrative domain will deploy a domain-wide auditor that is capable of monitoring all the transactions associated with the resources controlled by the domain. Our *ad hoc* trust inference model requires an auditing component that aims to (1) deter requesters from lying about their environment attributes, (2) catch inconsistencies between the self-claimed urgency level, and (3) propagate the auditing results back to identity and history attribute values.

The main task of the auditor is to monitor whether a requester lies about the urgency level associated with a request. Whenever there are major or minor crisis events [2], the information associated with the event including time, severity, and location, is given to the auditing service. The event's information will be used to map to a urgency level that will be then used to compare with the self-claimed urgency level associated with past transactions. In general, the auditing service only needs to check transactions whose urgency levels are relatively high to catch any inconsistencies. The auditing results will be folded back to our trust inference algorithm and be used to adjust the history attribute values.

Transaction History Currently our model considers the transaction history that contains only transaction of the owner. In order to consider also the transaction history done with other nodes, a reputation model may be utilized and the computation for trust value needs to be adjusted accordingly. In principle, more data on the transaction history of a requester will give higher accuracy in trustworthiness prediction. However in decentralized environments, it is infeasible to gather all the available transaction history from all possible sources. One simple approach is to have a collaborative filtering mechanism where several organizations form a trusted clique to share transaction histories of previous interactions. In the trust inference computation, additional attributes may be introduced to capture these factors. However, this may raise a privacy issue that is the access history of an individual may be traced and analyzed by clique members to infer additional knowledge, which would be impossible to obtain if the transaction histories are not shared. How to achieve privacy-preserving collaborative filtering in reputation systems remains an interesting open problem.

IV. Related Work

Most of existing research has extensively addressed the need of secure access, focusing on comprehensive policy designs and analysis and efficient management of users privileges and privacy. Recent studies found that in mission-critical systems (e.g., military, firefighting, or SCADA [12]) conventional access control mechanisms may be rigid for urgent information sharing scenarios and often fail to provide adequate supports for access in exceptional-yet-truly-needed situations [1], [5], [14], [7]. In critical infrastructures such as utility networks, oil and gas pipelines, and battlefield communications, there is an increasing need to secure the information collected from and about the infrastructure, and yet to be able to allow flexible data sharing to facilitate problem-solving.

Recently, there are a few notable papers proposing interesting solutions to the problem of flexible and controlled information sharing [1], [5], [14], [7], [15], [17], [18]. JASON report [5] presented a tokenized access framework and an economic model for regulating the tokens. In their proposed approach, tokens may be viewed as cash and can be spent to access sensitive information. A fuzzy multi-level security (MLS) model based on probability was proposed by Cheng *et al.* [1]. Despite the name, the work is not based on fuzzy logic, rather on a new probabilistic formulation of MLS model that supports quantified access decisions. Keppler, Swarup, and Jajodia developed a Flexible Authorization Framework (FAF) that redirects mission-related denied requests to corresponding entities who may serve as an override authority [7] and thus enables dynamic information sharing.

Compared to the existing above-described work, our approach gives a trust inference mechanism that (1) is based on a comprehensive profile of a requester, (2) utilizes the digital credential infrastructure, (3) adapts to environments, and (4) is rule-based.

Fuzzy logic has been used in many systems. Fuzzy logic system has been used to detect attacks in wireless networks including collision attacks, unfairness attacks, and exhaustion attacks [10]. The system observes the attack patterns and defines rules of responses. Most recently, a fuzzy logic system is developed for the trust management in grid computing environments [13]. The main goal of the work in [13] is to match the security policy of a grid computing server with a client based on the self-claimed security parameters. Our *ad hoc* trust inference model demonstrates a novel application of fuzzy logic in crisis information sharing. Furthermore, our work develops the model and architecture for building the profile of a requester by integrating multiple attributes associated with the request including the environment information.

Our trust model work is related to the existing work on recommendation or reputation systems in decentralized models [8]. Trust evidences that are generated by recommendations and past experiences have been used for trust establishment in both ad-hoc and ubiquitous computing environments [3], [11], [16], [2]. The work that is closest to ours is the parameterized authentication by Covington *et al* [2] that is built on subjective logic [6] to infer authentication decisions in pervasive environments based on incomplete, unreliable, and inaccurate sensor readings. Each sensor inputs a tuple representing belief, disbelief, and uncertainty with respect to the authentication of a person. The tuples from multiple sensors are aggregated using subjective logic. Subjective logic extends standard logic to use continuous uncertainty and belief parameters as opposed to discrete ones. Parameterized authentication explicitly defines and computes uncertainty values for each input. In comparison,

[2]We assume that this information is available to the public.

the uncertainty factors are implicitly captured in the fuzzy rule sets of our model. The inputs in our *ad hoc* trust inference are single crisp values. The advantage of this aspect is that it offers better interoperability with conventional frameworks where single-valued inputs are expected (as opposed to three-element tuples). Parameterized authentication uses the subjective logic operation called consensus to aggregate inputs. In comparison, we use aggregate and defuzzification in fuzzy logic. It remains an interesting open problem to compare the sensitivity of both approaches.

V. Conclusions and Future Work

We have described an *ad hoc* trust inference model where access decisions are adaptive to the identity, history, and environment associated with a request. Our trust inference model is built on fuzzy logic, which is simple to define. Most importantly, fuzzy logic system lends itself to a balanced and comprehensive decision making mechanism that mimics the process of human thinking. Using soft computing techniques is a promising direction for flexible and controlled information sharing. There are several exciting directions to pursue. For future work, we plan to study the sensitivities of fuzzy logic components on the trust score computation. For example, it is interesting to investigate and experiment various membership functions and fuzzy rule definitions in our model, in order to identify the impacts of parameter changes on the final decision making process. Another important problem to study is how to integrate the *ad hoc* trust inference system with predicate-logic based access control systems, in order to achieve smooth transitions between the two systems under normal and crisis situations. We would also like to explore how the model would withstand an attack by purposefully injecting wrong attribute values. We expect to use a combination of statistical and cryptographic techniques to address this issue.

Our sharing control mechanism can also be applied to protect personal data and used for digital identity protection. Recent studies and reports indicate that the sharing control mechanisms in current Web 2.0 environments are too weak and are unable to provide sufficient and effective privacy protections to the increasing number of users. Web 2.0 emphasizes on sharing and community-based collaboration. It is very different from the assumptions in the authorization models in conventional centralized or distributed systems where control and protection are their main goals. Imposing an overly strict information sharing control in Web 2.0 environments would defy the entire purpose of Web 2.0. In particular, in Web 2.0 an information provider should be able to share her contents with others who may be strangers to her. Therefore, a fundamental problem for realizing authorization in open systems such as Web 2.0 is for an information provider to evaluate the trustworthiness of unknown requesters. In addition, typical authorization and authentication systems are designed for system administrators and thus would be too complex to use by average users on a daily basis. Thus flexible and easy-to-use sharing control mechanisms have been lacking and such techniques will be extremely valuable for protecting user privacy in Web 2.0 environments. We plan to study the usability of *ad hoc* trust inference in protecting on-line personal information.

References

[1] 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 *SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pages 222–230, Washington, DC, USA, 2007. IEEE Computer Society.

[2] M. J. Covington, M. Ahamad, I. A. Essa, and H. Venkateswaran. Parameterized authentication. In *Proceedings of the 9th European Symposium on Research Computer Security (ESORICS)*, pages 276–292, 2004.

[3] L. Eschenauer, V. D. Gligor, and J. Baras. On trust establishment in mobile ad-hoc networks. In *Proceedings of the Security Protocols Workshop*, April 2002.

[4] J. L. Garnett and A. Kouzmin. Communicating throughout Katrina: Competing and complementary conceptual lenses on crisis communication. *Public Administration Review*, Special Issue, November-December:170 – 187, 2007.

[5] HORIZONTAL INTEGRATION: Broader access models for realizing information dominance. Technical report, MITRE Corporation, 2004. JASON Program Office. http://www.fas.org/irp/agency/dod/jason/classpol.pdf.

[6] A. Jøsang. Artificial reasoning with subjective logic. In *Proceedings of Australian Workshop on Commonsense Reasoning*, 1997. In conjunction with the Tenth Australian Joint Conference on Artificial Intelligence.

[7] D. Keppler, V. Swarup, and S. Jajodia. Redirection policies for mission-based information sharing. In *SACMAT '06: Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 210–218, New York, NY, USA, 2006. ACM.

[8] R. Kohlas and U. M. Maurer. Confidence valuation in a public-key infrastructure based on uncertain evidence. In *Proceedings of the Third International Workshop on Practice and Theory in Public Key Cryptography (PKC '00)*, volume 1751 of *Lecture Notes in Computer Science*, pages 93–112. Springer, 2000.

[9] T. Munakata and Y. Jani. Fuzzy systems: An overview. In *Communications of the ACM*, volume 37 (3), pages 69 – 76, 1994.

[10] Q. Ren and Q. Liang. Fuzzy logic-optimized secure media access control (FSMAC) protocol for wireless sensor networks. In *IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety (CIHSPS 2005)*, 2005.

[11] B. Shand, N. Dimmock, and J. Bacon. Trust for ubiquitous, transparent collaboration. *Wirel. Netw.*, 10(6):711–721, 2004.

[12] S. Singh, D. M. Nicol, and W. H. Sanders. Automatic verification of distributed and layered security policy implementation using the access policy tool. In *3rd Midwest Security Workshop (MSW)*, 2006.

[13] S. Song, K. Hwang, and Y.-K. Kwok. Trusted grid computing with security binding and trust integration. *J. Grid Comput.*, 3(1-2):53–73, 2005.

[14] V. Swarup, L. Seligman, and A. Rosenthal. Specifying data sharing agreements. In *POLICY 2006*, pages 157–162, 2006.

[15] R. Tamassia, D. Yao, and W. H. Winsborough. Role-based cascaded delegation. In *Proceedings of the ACM Symposium on Access Control Models and Technologies (SACMAT '04)*, pages 146 – 155. ACM Press, June 2004.

[16] G. Theodorakopoulos and J. S. Baras. Trust evaluation in ad-hoc networks. In *WiSe '04: Proceedings of the 2004 ACM workshop on Wireless security*, pages 1–10. ACM Press, 2004.

[17] D. Yao, K. B. Frikken, M. J. Atallah, and R. Tamassia. Point-based trust: Define how much privacy is worth. In *Proc. Int. Conf. on Information and Communications Security (ICICS)*, volume 4307 of *LNCS*, pages 190–209. Springer, 2006.

[18] D. Yao, R. Tamassia, and S. Proctor. On improving the performance of role-based cascaded delegation in ubiquitous computing. In *Proceedings of IEEE/CreateNet Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm '05)*, pages 157–168. IEEE Press, September 2005.

[19] L. A. Zadeh. Fuzzy logic, neural networks, and soft computing. In *Communications of the ACM*, volume 37 (3), pages 77 – 84, 1994.