Apply Measurable Risk to Strengthen Security of a Role-based Delegation supporting Workflow System

Weili Han\textsuperscript{1,2}, Qun Ni\textsuperscript{2}, Hong Chen\textsuperscript{2}
\textsuperscript{1}Software School, Fudan University, Shanghai, China
\textsuperscript{2}Department of Computer Science, Purdue University, West Lafayette, Indiana, USA
\texttt{wlhan@fudan.edu.cn, \{hanw, ni, chen131\}@cs.purdue.edu}

Abstract

Workflow systems often use delegation to enhance the flexibility of authorization. However, using delegation also weakens security because users may have difficulties understand and design correct delegation policies. In this paper, we propose the Measurable Risk Adaptive Role-based Delegation (MRARD) framework to address this problem. MRARD employs measurable risk for SSOs (System Security Officers) to provide a complementary protection mechanism in role-based delegation supporting workflow systems. In MRARD, when another enterprise user wants to use a delegated role to execute a task, a fuzzy logic based inference processor will infer the risk level. Based on simple risk adaptive decision policies, a decision module will determine whether the access should be granted under a certain risk mitigation action.

1. Introduction

The application of measurable risk is recently recognized as an effective approach to access control in dynamic and emergent application scenarios\cite{1}\cite{2}\cite{3}. In this paper, we propose a novel framework, called Measurable Risk Adaptive Role-based Delegation (MRARD), to strengthen the security of a role-based delegation supporting workflow system.

Delegation\cite{4}\cite{5} is a flexible mechanism in an enterprise, which ensures that business processes are executable when a key user temporarily leaves his/her position. However, a delegation may assign a security administrative task to an enterprise user who is not well-trained in security administration. Thus, it is difficult for the enterprise user who is not the SSO to understand and design traditional delegation policies.

Although SSOs can define high level policies and rules to avoid low-level mistakes committed by a delegator\cite{4}\cite{5}, the traditional access control model\cite{6} and the delegation model\cite{4} are still too rigid in dealing with risky delegations. In traditional methods of role-based access control and role-based delegation, every one has the same priority when he/she has qualified roles or attributes according to the delegation policies. Finer-grained rules and policies can reduce the probability of risky delegations, but are hard to be well-defined due to their complexity and size. The SSOs are, therefore, falling into a dilemma where they must trade off between keeping the business processes executable and enforcing the Principle of Least Privilege. On the one hand, if the SSOs make less strict rules and policies which allow more users to accept delegation, the delegator might maliciously or accidentally assign a role to an unsuitable user when the SSOs are unaware until a serious consequence happens. On the other hand, if the SSOs make finer rules and policies to limit who can receive delegation, the workflow instance might not be executable due to absence of a key user.

Now, a measurable risk adaptive method provides an alternative and flexible way to protect an information system by identifying risk and mitigating risk \cite{1}\cite{2}\cite{7}. However, current research works about measurable (or quantitative) risk\cite{1}\cite{7} can not be applied directly to an enterprise-oriented role based application.

This paper, therefore, proposes a novel access control framework named MRARD for a role-based delegation supporting workflow system. MRARD explicitly applies measurable risk as an indicator to strengthen security of a role-based delegation supporting workflow system. When a user wants to use a delegated role to execute a task, the system will evaluate the risk according to relevant fuzzy risk evaluation policies which are predefined by the SSOs. After risk evaluation, the system can determine whether the execution is allowed according to risk-adaptive decision policies, the risk and a risk mitigation action.

The contributions of this paper are as follows:

- We propose a novel measurable risk adaptive framework, MRARD, which explicitly evaluates the risk level in a role-based delegation supporting workflow system, and strengthens the security of role-based delegation;
- We leverage fuzzy set and fuzzy logic to deal with the multi-value problem which is a major challenge during evaluating risk level.

The rest of the paper is organized as follows: section 2 introduces the framework; section 3 introduces the background knowledge; section 4 introduces MRARD; section 5...
leverages fuzzy set to define a formal fuzzy risk mitigation language, and introduces how to evaluate risk level by the theory of fuzzy logic. In section 5, we also introduce a simulated case which explains how the risk evaluation works; section 6 describes how to make a risk adaptive decision; section 7 analyzes the efficiency of MRARD; section 8 discusses the related works; the final section summarizes the paper and discusses future work.

2. A Risk Adaptive Access Control Framework

We follow the three guiding principles in JASON Report[2] to design a measurable risk framework that evaluates risk level and determines whether the risk level is acceptable under a certain risk mitigation action. The framework includes four steps:

- **Risk Components Measurement**: SSOs specify which risk components are the main ones, and specify their respective measurement method;
- **Risk Level Evaluation**: the system will evaluate how much risk there is according to the previous specified risk components. This step is generally complex because it aggregates the risk components which are not conditionally independent, but connected with each other. It is hard to use a simple formula to get a precise value. But we use ten different risk levels from risk level 0 to risk level 9, instead of the precise value.
- **Define policies of acceptable risk level**: SSOs define which risk level can be acceptable when the delegation happens. The policies can either be applied to all delegation in a system, or depend on concrete delegation respectively.
- **Determine whether the risk level is acceptable under a certain risk mitigation action**: A risk aware request is allowed according to evaluated risk level and risk mitigation action. A risk mitigation action could reduce the degree of risk level.

3. Background

3.1. Risk Measurement

Risk is defined as the expected value of damages[7] or the possibility of loss or injury in the Merriam-Webster dictionary[1]. But due to the complexity of risk, how to measure risk is variable in different domains. For example, Value at Risk (VaR)[8] was defined as a specified probability and a specified time horizon for a specific portfolio of financial assets. VaR is a risk measurement based on historic data. However, in the access control field, there are not enough log data that can be used to measure the risk[8].

Therefore, research in the access control field started to explore risk measurement recently. Cheng et al. [1] proposed a risk measurement method based on temptation and inadvertent disclosure. And Cheng et al. tried to use both the attributes (security level and category) of subjects and objects to calculate the probability of expected loss. However, in major commercial information systems, the attributes security level and category are usually absent, and the access control policies are more complex than those in a Multi-Level Security system.

This paper leverages fuzzy set and fuzzy logic to evaluate final risk level rather than calculate a precise value from a formula[1] or a Bayesian network[10].

3.2. Role-Based Access Control and Role-Based Delegation

Role-based access control is a popular access control model in the recent decades[6]. A comprehensive introduction of general Hierarchy Role-Based Access Control model is given in[6]. Figure 2 shows an instance of role hierarchy and a user-role configuration.

Role-based delegation was defined as a process whereby an active entity in a distributed environment authorizes another entity to access resources[4]. Thus, Li et al.[5] defined a trust-management method to deal with the delegation in a distributed environment. And Zhang et al.[4] defined a role-based delegation and revocation model to manage role-based delegation.

3.3. Workflow System

Delegation discussed in this paper is combined with a workflow system[11], because a workflow system is a
4. Risk Vector in MRARD

Definition 3: Measurable Risk

\[ \text{Measurable Risk} = (\text{Rank Diff}, \text{Range Diff}, \text{Workflow Value}) \]

In MRARD, we define measurable risk as a vector where each component is measurable. Each measurable risk vector consists of three components: Rank Diff, Range Diff, Workflow Value.

Risk in delegation partly arises when a user wants to use a delegated powerful role to execute a task. And this part of risk includes two components: Rank Diff, Range Diff. Because a powerful role generally has some sub-roles, the user could activate different sub-roles (we note the executor’s role as an active role) to execute different tasks. As a result, different active roles will bring up different risks:

- **Rank Diff**: Rank Diff is measured by Rank Difference between an active role and a user. The ranks of roles come from SSO’s assignment, when the ranks of users are derived from ranks of roles. As is shown in Figure 2, a structure of role hierarchy actually is a two-dimension diagram. In the vertical direction, there is an implicit rank which is similar to the security level in a Multi-Level Security system. For example, in Figure 2, the rank of CARDDIR is higher than the rank of any other role which is directly below CARDDIR, such as CS;

- **Range Diff**: Range Diff is measured by permission difference between an active role and a user. In the horizontal direction of the role hierarchy structure, there is a measurable range according to access control permissions. This can resolve the problem where the measurable Rank Diff is negligible, but the delegatee is not a member of the active role. For example, in Figure 1, a user (Cathy) of ORDIR wants to activate the delegated CARDDIR to execute a task. And the Rank Diff between Cathy and CARDDIR is negligible. However, because the two roles are defined in two different studios, the delegation obviously has risk due to their difference of permissions between CARDDIR and ORDIR.

The third component of risk is Workflow Value. It is measured by importance degree of an object (an workflow instance in MRARD). Generally, as is shown in Definition 1, Workflow Value requires SSOs’ assignment before a workflow instance is started.

4.1. Measure Rank Diff

This component of measurable risk is calculated by the difference between Role’s Rank of an active role and User’s Rank. The former describes the basic rank of an active role,
and the latter describes the user’s rank which is derived from Role's Rank by the relationship of UA (User-Role Assignment).

The function of Role's Rank is defined as follows:

**Definition 4: Role’s Rank**

\[ R(r : R) : R \rightarrow N \]

Here, \( R(r : R) \) returns Role’s Rank (\( R(r) \)) of an input role. \( R(r) \) is defined by a SSO after a role hierarchy is set up. And \( R(r) \) is a number which meets the following condition:

\[ \forall r_1, r_2 \in R : r_1 \succeq r_2 \Rightarrow R(r_1) \geq R(r_2) \]

Here, \( r_1 \succeq r_2 \) means \( r_1 \) is a directly super role of \( r_2 \).

**Definition 5: User's Rank**

\[ R(u : U) = \max\{R(r : R)|r \in \text{assigned_roles}(u)\} \]

Here, \( U \) refers to a user set; \( \text{assigned_roles}(u) \) refers to all roles assigned to the user. \( R(u) \) is derived from the maximum Role’s Rank of all roles of a user.

Based on Definition 4 and Definition 5, we define IDX_{rank\_diff} as follows:

**Definition 6: IDX_{rank\_diff}**

\[ IDX_{rank\_diff}(u : U, ar : R) = \begin{cases} \frac{a^{(\text{assigned\_permissions}(ar) - \text{assigned\_permissions}(u))} - 1}{\text{upper} - \text{assigned\_permissions}(ar)}, & \text{if } R(ar) \geq R(u) \\ 0, & \text{if } R(ar) < R(u) \end{cases} \]

Here, \( a \) is a real number that is greater than 1, and \( \text{upper} \) is a real number that is greater than the maximum allowed value of \( R(ar) \).

IDX_{rank\_diff} refers to the direct difference between the User’s Rank and the activated Role’s Rank. The formula satisfies the following properties:

- \( IDX_{rank\_diff} \geq 0 \); and \( IDX_{rank\_diff} = 0 \), when the User’s Rank is larger than the activated Role’s Rank;
- \( IDX_{rank\_diff} \) increases as the activated delegated Role’s Rank increases, and is biased toward the higher Role’s Rank.

**Definition 7: Range_Diff**

\[ \text{Range\_Diff}(u : U, ar : R) = \begin{cases} \frac{1}{1 + e^{-(1 + k \times IDX_{rank\_diff}) \times m}}, & \text{if } IDX_{rank\_diff} > 0 \\ 0, & \text{if } IDX_{rank\_diff} = 0 \end{cases} \]

We choose sigmoid function[1] to calculate Range_Diff. In Definition 7, \( ar \) refers to an active role; \( k \) determines the slope of the Range_Diff curve with regard to IDX_{rank\_diff}; \( m \) is equal to IDX_{rank\_diff} when Range_Diff(\( u, ar \)) = 0.5.

Range_Diff is a probability that User’s Rank reaches the activated Role’s Rank. \( \text{range\_diff} \) monotonically increase with IDX_{rank\_diff}. And if the User’s Rank is equal to, or larger than the Role’s Rank, \( \text{range\_diff} \) should be negligible and represented by zero.

### 4.2. Measure Range_Diff

**Range_Diff** measures the risk component when an active role is delegated to a user whose assigned roles exclude the active role, but the User’s Rank is the same as or similar to the Role’s Rank.

Definition of Range_Diff is based on the following assumption: If a user has fewer permissions an active role is assigned, risk is higher. The assumption comes from the following fact: When the delegator and the delegatee are at the same division or involved in similar jobs, the risk is low because the delegatee is familiar with the delegated role’s job function. Therefore, the intersection of delegatee’s and delegated role’s permissions is big. However, when the delegator and the delegatee are at different divisions, even different companies, the risk is high because the delegatee is not familiar with the delegated role’s job function. Therefore, the intersection of delegatee’s and the active role’s permissions is small. We formally define the Range_Diff as follows:

**Definition 8: Range_Diff**

\[ Range_{\text{Diff}}(u : U, ar : R) = \begin{cases} \frac{1}{1 + e^{-(1 + k' \timesIDX_{range\_diff}) \times m'}}, & \text{if } IDX_{range\_diff} > 0 \\ 0, & \text{if } IDX_{range\_diff} = 0 \end{cases} \]

We also choose sigmoid function to calculate Range_Diff. In Definition 9, \( ar \) refers to an active role; \( k' \) determines the slope of the Range_Diff curve with regard to IDX_{range\_diff}; \( m' \) is equal to IDX_{range\_diff} when Range_Diff(u, ar) = 0.5.

### 4.3. Measure Workflow_Value

WorkflowImport has been defined in Definition 1. When a user activates his delegated role or one of its sub-roles to
execute a workflow instance, there is more risk when the workflow instance has more Workflow_Value.

5. Risk_Level Evaluation

5.1. Fuzzy Set and Fuzzy Logic

A fuzzy subset FS of a set S can be defined as a set of ordered pairs, each with the first element from S (name of subset), and the second element from the interval [0,1] (membership), with exactly one ordered pair present for each element of S. A membership function of the fuzzy set describes a mapping between FS and membership. Fuzzy logic was introduced by Dr. Lotfi Zadeh[13]. It is a superset of Boolean logic. Fuzzy logic is extended to handle very large scale and complex scenarios in the real world. Therefore, they are applied in many decision-based control applications. A fuzzy expert system is one of the typical applications. And its general inference process includes four steps: FUZZIFICATION; INference; COMPOSITION; DEFUZZIFICATION.

5.2. Fuzzy Risk Evaluation Policy

The process of Fuzzy Risk Evaluation evaluates risk_level according to input measurable risk which is defined in Definition 3.

Definition 10: Fuzzy Risk Evaluation Policy

\[ FREPolicy = \langle FS_{Rank\_Diff}, FS_{Range\_Diff}, FS_{Workflow\_Value}, FS_{Risk\_Level} \rangle \]

Here, FS is a prefix notation for fuzzy subsets, which include, for example, high, middle and low. Each fuzzy subset is generally defined by a membership function.

The membership functions of fuzzy subsets are valuable to further study, but they are not the main work of this paper. We simply define a fuzzy subset as a range attribute, such as (lowbound, highbound). The range refers to the fuzzy areas of low and high, and non-zero area such as middle. We also define the membership functions as:

\[
\begin{align*}
\text{middle}(x) &= \\
&= \left\{ \\
&0, \text{if } x \leq \text{lowbound} \\
&\frac{(x-\text{lowbound})}{\text{highbound}-\text{lowbound}}, \text{if } \text{lowbound} \leq x \leq \text{highbound} \\
&100\% \text{, if } x \geq \text{highbound} \\
\end{align*}
\]

\[
\begin{align*}
\text{high}(x) &= \\
&= \left\{ \\
&0, \text{if } x \leq \text{lowbound} \\
&\frac{x-\text{lowbound}}{\text{highbound}-\text{lowbound}}, \text{if } \text{lowbound} \leq x \leq \text{highbound} \\
&100\% \text{, if } x \geq \text{highbound} \\
\end{align*}
\]

For example, we define the membership functions of fuzzy subsets for Rank_Diff: high, middle, low. And we define the fuzzy area of high is (40%, 50%); the range of middle is (0%, 50%); and the fuzzy area of low is (0%, 10%).

Here is an example policy of Definition 10: <high, high, high, high>. It means when a user executes a task, if the Rank_Diff is high, the Range_Diff is high and the Workflow_Value is high, the risk_level is high.

It is not difficult to design Fuzzy Risk Evaluation Policies, because the number of Fuzzy Risk Evaluation Policies is usually limited, the policy is intuitionistic and there is a case-based method to assist the policy checking. First, the number of the policies is determined by the number of the risk’s components and the number of their individual fuzzy subsets in each Fuzzy Risk Evaluation Policy. Because every scenario with measurable risk should belong to one fuzzy subset of risk_level, there is at most \(3^3 = 27\) policies that SSOs should make. In addition, a Fuzzy Risk Evaluation Policy has an important feature: Monotone Non-decreasing (which can help SSOs when they are making policies). That is,

\[ \frac{\partial FS_{Risk\_Level}}{\partial \text{component}} \geq 0 \]

Here, the component refers to each component except for FS_{Risk\_Level} in the Fuzzy Risk Evaluation Policy definition. Because of Monotone Non-decreasing, it is impossible for some potential policies to co-exist based on the feature. For example, it is impossible for the policy of <high, middle, high, high> to coexist with the policy of <high, high, high, middle>.

Next, the Fuzzy Risk Evaluation Policy is intuitionistic for SSOs, because the Fuzzy Risk Evaluation Policy uses intuitionistic words like high, low, etc. to express the degree of the components.

Last but not least, we can design a case-based method to check inconsistency of Fuzzy Risk Evaluation Policies. During the case-based checking, the SSOs use a typical scenario which should belong to a certain risk_level. For
5.3. Evaluation Process based on Fuzzy Logic

Based on the policy definition in the above section, we use defined policies to determine risk_level for a measurable risk vector.

As is shown in Figure 3, the process of evaluation for a measurable risk vector includes four standard steps: Fuzzification; Inference; Composition; and Defuzzification.

1) Fuzzification will use the membership functions of fuzzy subsets for each component of measurable risk to assign a degree to every fuzzy subset;

2) Inference will evaluate a measurable risk vector by each Fuzzy Risk Evaluation Policy. Because the policy uses the AND operator in the Fuzzy Risk Evaluation Policy definition, we use MIN inferencing to get the evaluation result of each policy;

3) Composition will compose all the results from the Inference phase. Due to MIN inferencing, we use MAX composing to compose the results;

4) Defuzzification will use the membership function to defuzzificate the result from the Composition phase. We use CENTROID rather than MAXIMUM to get the final risk_level.

5.4. An Simulated Case Study

As is shown in Table 1, we describe the parameters of membership functions of fuzzy subsets for the four components. And as is shown in Figure 3, we describe three Fuzzy Risk Evaluation Policies. Then we will demonstrate the calculation process when we assume we get a vector of measure_risk as: ⟨45%, 45%, 75%⟩.

We use MIN-MAX to do inference. According to the first policy in Figure 3, the result of inference is 19% of the high fuzzy subset of Risk_Level. The inference result of the second policy is 8% of the middle fuzzy subset of Risk_Level. And the result of the third policy is 0% of the low fuzzy subset of Risk_Level.

Through composition and defuzzification, we calculate that the Risk_Level is 6. But, if we are given the risk of ⟨5%, 45%, 85%⟩, the risk_level is zero after inference, composition and defuzzification according to the membership functions in Table 1 and the policies in Figure 3. That is unreasonable. The reason is the policy set is incomplete or not well designed. Thus, we should improve the policy set by designing new Fuzzy Risk Evaluation Policies.

5.5. Problems in Fuzzy Risk Evaluation

5.5.1. Other Components of Risk and Multi-Level Inference. An information system is so complex that there are a few other components of risk. For example, credit of a user, which is often used in a financial system, also affects the risk in role-based delegation. A user with poor credit could maliciously operate his/her delegated role. The potential operation could lead to more risk. Thus, further research will explore more risk components in the measurable risk model.

A multi-level structure for fuzzy inference is an alternative way in dealing with more components in measurable risk. For example, Figure 4 shows a multi-level structure for fuzzy inference for measurable risk evaluation. In the multi-level structure, the risk inference is divided into two levels: the processor in the first level infers delegation risk, and the processor in the second level infers final risk_level. This design can limit every inference processor to no more than 3² = 9 policies.

5.5.2. Define Membership functions of fuzzy subsets and Fuzzy Risk Evaluation Policy Set. To design well-defined membership functions of fuzzy sub sets and a Fuzzy Risk Evaluation Policy set is an important issue in MRARD, although the number of the membership functions and the policies is limited. The analysis in section 5.2 preliminarily gives us a potential method which uses real typical cases to check Fuzzy Risk Evaluation Policies. The future work will research how to simplify to the design of these membership functions and policies.
6. Determine an Access Request

6.1. Risk Mitigation

The decision in MRARD depends on both risk_level and a risk mitigation action. We define Risk Mitigation action as follows:

**Definition 11: Risk Mitigation**

$$\text{Risk}_\text{Mitigation} = \text{Actions} \times \text{Risk}_\text{Level} \times \text{MitigationEffect}$$

Definition 11 means on which risk_level a risk mitigation action can be effective and how much effect the mitigation has. In Definition 11, Actions refers to all real risk mitigation actions, for example, "send an email to the SSOs board" or "cooperate by another user with a delegatee’s highest-rank role which is assigned by SSOs"; Risk_Level refers to risk_level at which the risk mitigation action can work; and MitigationEffect refers to how much effect the risk mitigation action can produce, for example, "reduce 50%".

Cheng et al.[14] bound risk mitigation with a certain risk_level. But in our framework, some risk mitigation actions can independently work on a certain risk_level, MRARD will select the most efficient and valid risk mitigation action to implement risk_level according to the defined risk mitigation actions.

6.2. Decision Policies and Determine a Request

The decision polices are simple and variable. That is, a SSO can pre-define acceptable risk_level for critical tasks. In addition, a SSO can also define a policy that accepts any one of the three users of the lowest risk_level under a certain risk mitigation action.

MRARD can infer risk_level according to context (Rank_Diff, Range_Diff, Workflow_Value) of a request and a risk mitigation action. And based the above risk adaptive decision policies, MRARD can dynamically determine whether the request can be allowed under the risk mitigation action.

7. Efficiency of MRARD

MRARD can strengthen security in a workflow system which supports role based delegation. MRARD is a complementary mechanism to protect sensitive data in a role based delegation supporting workflow system. In front of MRARD, the delegation policies designed by professional SSOs will provide the workflow system with the hard boundary[1] protection. Furthermore MRARD explicitly measures the potential risk which the traditional mechanism can not provide. MRARD can efficiently identify risk according to task execution context (Rank_Diff, Range_Diff and Workflow_Value). And the system can make a dynamical decision due to different situations in delegation. Thus, MRARD provides a more flexible security complementary mechanism for a role-based delegation supporting workflow system.

8. Related Works

Researchers have developed some languages and methods to specify who can delegate which role to a certain user[5][4] and how to restrict the delegatee. But they could be too rigid in dealing with the role-based delegation. In these researches all users have the same priority to accept delegation if the policies and rules allow, and the delegatee can apply his/her delegated role to any workflow instance only if the delegated role is qualified for workflow designs. As a result, the SSOs can not deal with the potential risk when the workflow system and the users are drastically changing. This paper explicitly applies the measurable risk to deal with the risk that comes from delegation.

Risk is argued as a more flexible way to deal with a dynamic information system[2][1][15]. However, currently, many researches focus on risk in a traditional financial system[8], an information system in an enterprise[9][10]. Even though researchers have finished some newest researches on access control systems[16][1][7][15], there are some limitations in these researches as are analyzed in section 3.1, when these researches are applied to a role-based enterprise-oriented application.
Dimmock et al. [16] used risk thresholding predicates to deal with the multi-value problem of risk and cost in a distributed file storage and publication service. However, the predicates are not good at resolving the multi-value problem. Zhang [3] discussed a risk adaptive method which considered risk and benefit. Different from the papers of [16] [3], our paper proposes a measurable risk as a vector which consists of measurable risk components, applies the theory of fuzzy set to define the Fuzzy Risk Evaluation Policy.

9. Conclusions and Future Work

This paper proposes a framework which explicitly evaluates risk level and decides a request according to the risk level and a risk mitigation action. Then, the paper proposes a novel framework, MRARD, that applies measurable risk to strengthen security in a role-based delegation supporting workflow system. MRARD explores a novel way to strengthen the security. When a user wants to execute a task in a workflow instance, the system will explicitly evaluate the potential risk components. In MRARD, we leverage fuzzy set and fuzzy logic to resolve the multi-value problem in evaluating the risk level. Different from rigid binary-decision of traditional access control models, MRARD can evaluate the risk level of each delegation, and determine dynamically whether a risk-aware action can be allowed under a risk mitigation action.

Further research will design a general framework which trades off risk and benefit that come from a risk-aware action.

Acknowledgement

Thank Prof. Ninghui Li, Prof. Min Li for their contributions to our paper. Thank Anonymous Reviewers for their comments.

References


