RC24190 (W0702-085) February 20, 2007
Computer Science

IBM Research Report

Fuzzy Multi-Level Security: An Experiment on Quantified Risk-Adaptive Access Control

Pau-Chen Cheng, Pankaj Rohatgi, Claudia Keser, Paul A. Karger
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Grant M. Wagner, Angela Schuett Reninger
U.S. Department of Defense

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Fuzzy Multi–Level Security : An Experiment on Quantified Risk–Adaptive Access Control*

Pau–Chen Cheng *pau@us.ibm.com*
Pankaj Rohatgi *rohatgi@us.ibm.com*
Claudia Keser *ckeser@us.ibm.com*
Paul A. Karger *karger@watson.ibm.com*
IBM Thomas J. Watson Research Center

Grant M. Wagner
Angela Schuett Reninger
{*gmw,amschue*}*@tycho.ncsc.mil*
US Department of Defense

Abstract

The goal of this paper is to present a new model for, or rather a new way of thinking of adaptive, risk–based access control. Our basic premise is that there is always inherent uncertainty in access control decisions and such uncertainty leads to unpredictable risk that should be addressed in an explicit way. Many different access control models have been studied and practiced extensively. We choose to expand the well–known, Bell–Lapadula model based Multi–Level Security (MLS) access control model as a proof–of–concept case study for our basic premise. The resulting access control model is more like a Fuzzy Logic control system [Jyh97] than a traditional access control system and hence the name "Fuzzy MLS". A short, but more recent version of this article is to appear in the 2007 IEEE Symposium on Security and Privacy.

1 Introduction

Our work is motivated by the fact that many organizations, especially those in the national security and intelligence arena, are unable to rapidly process, share and disseminate large amounts of sensitive information in order to support informed decision making to rapidly respond to external events. A major inhibitor is the inflexibility of current access control models to deal with such dynamic environments and needs. For example, consider a complex organization in this arena with multiple, hierarchically organized departments, each holding information and data, understanding the significance of isolated events and formulating an effective response may require users and management in the organization to pool together information available within multiple departments (i.e., to *connect the dots*). Clearly, the information that needs to be pooled together would depend on the external event and the analysis approach adopted, and this *cannot be predicted in advance*. Traditional access control policies based on roles aligned with the organization chart can degrade the effectiveness of the response or may force the organization to take undue risk; either the policy is too rigid and does not allow necessary information to be shared or it is forced to give users near–blanket access rights which results in unaccountable risk of information leakage. Studies such as the JASON Report [JPO04] were explicitly commissioned to investigate barriers to information sharing and have reached a similar conclusion

*This Research is continuing through participation in the International Technology Alliance sponsored by the U.S. Army Research Laboratory and U.K. Ministry of Defense.

that existing security policy models are too rigid and do not allow necessary information to be shared; as a reaction some organizations have even set up complex mixes of loose and ad–hoc policies that may result in unaccountable risk of information leakage. In some cases, a culture has developed along the line of the old saying "it is better to ask for forgiveness rather than for permission". The problem is due to the fact that existing access control policies specify access decisions *statically* whereas the environments in which the policies are applied change *dynamically*. Thus the ideal case where an organization continually optimizes access control based on risk vs. benefit tradeoffs while capping overall risk cannot be realized.

Our work is geared towards creating and validating a novel, risk and information flow based access control model that can revolutionize the way work is conducted in such organizations. In particular, we will show how the scenario above can be solved by making access control much more dynamic and flexible using a risk management approach based on quantified risk estimates. Essentially, our Fuzzy MLS model quantifies the risk associated with an access and can even allow risky information flows needed by a user provided the risk can be accounted for and controlled. The eventual goal is to create a system that encourages information sharing and prudent risk-taking behavior among its users to maximize the benefit to the organization while at the same time keeping users accountable for their actions and capping the expected damage an organization could suffer due to sensitive information disclosure. In addition an organization will be able to control risky information flows dynamically based on its current operational needs, risk tolerance and environment.

This paper is organized in the following way: section 2 discusses the general idea of quantified risk–adaptive access control, section 3 discusses risk vs. benefit tradeoff, section 4 discusses related work, section 5 presents the Fuzzy MLS model, section 6 describes the prototype implementation of Fuzzy MLS , section 7 discusses on–going and future research, and section 8 concludes.

2 Quantified Risk–Adaptive Access Control (QRAAC)

This section discusses our general idea of *Quantified Risk–Adaptive Access Control* (*QRAAC*). We will first discuss our intuitive interpretation of risk and its relationship to access control, and then expand the discussion into QRAAC.

2.1 Risk in Access Control Decisions

The Merrian–Webster dictionary defines the word *risk* as "*the possibility of loss or injury*". Likewise, we intuitively interpret risk as *the likelihood that some event happens in the future to incur some damage*. The key words are "likelihood", "future" and "damage". Risk is always about uncertainty in the future; and there will be no risk if there will be no damage. The reason an organization controls access to its information is that it cares about the potential damage caused by possible abuse of the information. Since *the future is unpredictable*, the only way to be absolutely sure that there will be no damage is to deny any access to any information that is deemed sensitive or valuable. This is not a practically viable option; for an organization to achieve its objectives, it needs to allow some access to its information by its employees or even its partners so they can do their jobs. So the organization always has to accept certain level of risk associated with information access as a cost of doing its business. To reduce such risk, it is a common practice, especially in the national security and intelligence arena, to require a person to go through a background investigation before he/she is granted a security clearance to access classified information. However, such a clearance is no guarantee for future behaviors. In reality, most leaks of classified information are done by people with high security clearance. In this sense, an access control policy could be considered a *assessment of the future* that tries to balance the future risks with the future needs. Such an assessment is inevitably *imprecise and incomplete* due to the unpredictability of the future. Therefore every access control decision made according to a policy is at best an *educated guess* of the future and is associated with certain risk. In practice, there will always be risk vs. need tradeoffs that are not forseen by the policy, even in policies which allow for pre–specified exceptions. These unforseen tradeoffs

often result in the *creation of ad–hoc exceptions* outside the policy in order to meet practical needs [JPO04]. These ad–hoc exceptions often need time–consuming human approvals and their associated risk is hard to account for. An excessive usage of ad–hoc exceptions could result in *unbounded risk* and *defeat the very purpose of having an access control policy*.

It would be nice to bring these *unforseen*, ad–hoc exceptions into the access control model so that these exceptions can be granted in a timely manner and their associated risk is accounted for. This would require a computer system to know when to bend the policy to grant an exception, and the system has to know *how much* the rule would be bent. Otherwise it could imply unlimited bending and abandoning the access control policy.

QRAAC is meant to answer this "how much bending" question; it goes further and enables the system to take proportional risk mitigation measures[1] to account for and reduce the risk.

2.2 Adaptive Access Control Using Quantified Risk Estimate

In this section we will define quantified risk, then present the basic idea of implementing adaptive access control using quantified risk.

We define quantified risk as the *expected value of damage*.

$$quantified\ risk\ =\ (probability\ of\ damage)\ \times\ (value\ of\ damage) \tag{1}$$

The *probability of damage* is the chance that an event happens to incur the damage. The *value of damage* is a quantified measurement of the damage. We do not define the unit of "value" but consider it is the job of a policy writer to determine the proper unit for his/her particular context. Quantifying risk means determining the probability and the value. Due to the *unpredictability of the future*, neither the probability nor the value can be determined with absolute certainty. The best one can do is to produce good *estimates* of them to compute good *quantified risk estimates*. Using risk estimates to make decisions is common practice in many fields; a good example would be how insurance premiums are determined.

We propose that the existing static access control models with binary "allow/deny" decisions be replaced by a dynamic, multi–decision access control model based on quantified risk estimates and risk tolerance. This model is shown in Figure 1 where the *risk scale* represents the range of quantified risk estimates which is further divided into multiple *bands of risk*. The quantified risk estimate for any access falls into one of these risk bands. Each band is associated with *a decision and an action*; *the decision, the action and band boundaries are all determined according to risk tolerance* and *can be changed when risk tolerance changes*. The top band would be associated with the decision "deny" because the risk is too high; we call the lower bound of the top band *hard boundary*. The bottom band would be associated with the decision "allow" because the risk is low enough; we call the upper bound of the bottom band *soft boundary*. A band between the hard and the soft boundaries can be associated with the decision *allow with risk mitigation measures* which are actions such as increased auditing, application sandboxing [CC03], charging the risk to the user, etc. In summary, Quantified Risk–Adaptive Access Control has the following main characteristics:

- An access control policy is defined within the framework of *a scale of quantified risk estimates*.
- The decision model is a *non–binary, multiple–choice decision model* with *risk tolerance* and *risk mitigation* taken into account.
- An access control policy is *adaptive to changing risk tolerance*.

2.3 Computing Quantified Risk Estimate

This section discusses our general idea of computing quantified risk estimate. Section 5 will give a detailed discussion on computing quantified risk in the Fuzzy MLS model.

[1] More details in Appendix B

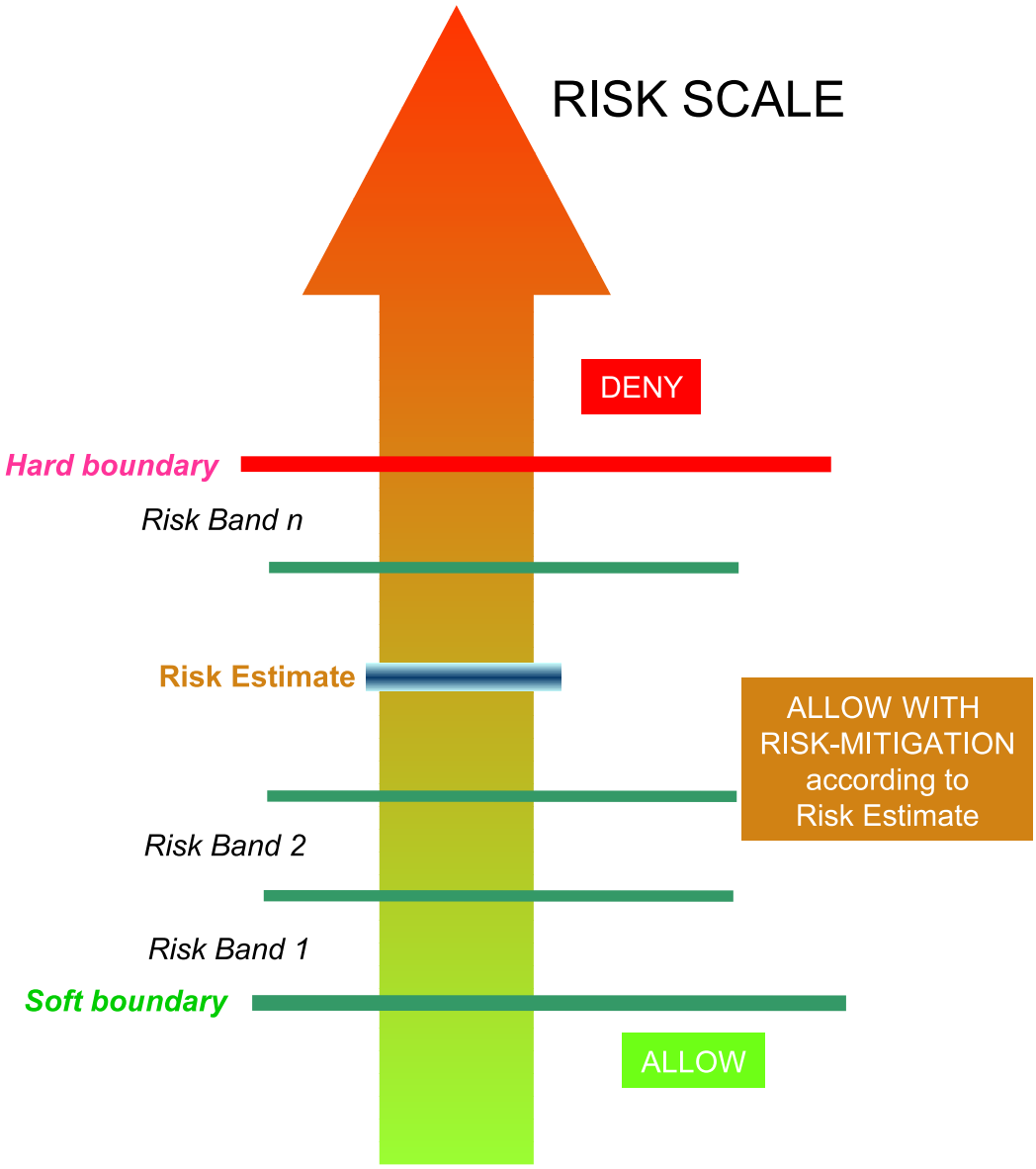

Figure 1: Risk–Adaptive Access Control on a Risk Scale

To compute a quantified risk estimate, we need to derive the *probability* and *value* in formula 1. Our research has been focused on deriving the probability. We do believe that any organization practicing risk management or MLS access control has a procedure in place to assess, or at least to classify the values of information. These classifications can be used to derive the corresponding value. For example, the DoD sensitivity levels are already defined qualitatively based on potential damage: *Top-Secret* is reserved for information whose disclosure can result in exceptionally grave damage and *Secret* is reserved for information whose disclose will result in serious damage [DoD97].

To derive the probability, we observed that while the true probability cannot be derived accurately in general, it is feasible to make a qualitative comparison between two accesses based on the *likelihood that the accessed resource will be misused*. If such a comparison is consistent, meaning that if access A is deemed more likely to result in misuse than access B and access B is deemed more likely to result in misuse than access C, then A is always deemed more likely to result in misuse than C, then one can put this comparison into a formula to compute *risk indices* which are *relative comparisons* of the likelihood. The higher the index, the higher

the perceived likelihood of misuse. *Probabilities can be assigned to risk indices* in a way that is commensurate with experience, intuition and threat assessment. The only requirement is that the probabilities are monotonically increasing with respect to risk indices. Of course, such probability assignment is a guess, but we would submit that *All access control policies and decisions are guesses* as discussed in section 2.1. In other words, *QRAAC brings the "guess" nature of access control up–front and deals with it explicitly.*

It is possible to skip the *risk–index* step and to directly assign a probability to an access. However, our experience with the Fuzzy MLS work taught us the value of using risk indices:

- Since a probability has to be in the range [0, 1] but the range of risk indices is not constrained, risk indices offer a much better *resolution* to encode the intuition behind the qualitative risk comparison.
- The assignments of probabilities to risk indices is a mapping from the qualitative risk comparison to probabilities. This mapping is based on intuition and experience and should be fine tuned over time, but the encoding of the risk comparison, namely the risk indices, can be kept fixed and these fixed indices would likely make the fine tuning easier.

In practice, many factors are contributing to risk and it may be too difficult to design one risk index formula that includes all the factors. Furthermore, even if such a formula could be designed, it is likely that the formula will contain many tunable parameters and it may be too difficult for a security administrator to comprehend and maintain such a formula. Our experience with Fuzzy MLS taught us that this problem can be addressed by the usual "divide and conquer" approach. For each risk contributing factor, a formula can be designed to compute the risk indices for that factor and probability assignment can be defined for these indices. The set of all factors can be partitioned into small subsets such that the relationship among the factors in a subset is better understood and the joint probabilities of these factors can be computed. Then these subsets are considered to be independent of one another and their joint probability can be computed.

3 Risk vs. Benefit Trade–off

A primary cause of an access control policy being subverted is that the policy conflicts with individual users' legitimate needs. The QRAAC model addresses this issue by allowing some risk taking when the risk of an access is between the hard and the soft boundaries. An organization's optimal goal should be encouraging prudent, calculated risk taking by users to achieve better results while still keeping the overall risk within the organization's risk tolerance, without micro-managing the human users. This optimal goal can be achieved in different ways based on how an organization chooses to influence its user behaviors.

One such system that we propose is similar to a credit card system. Each human user will be given a risk budget as a *line of risk credit* in some units of risk. If a user makes an access whose risk is between the soft and the hard boundaries, then the difference between the risk and the soft boundary (in units of risk) will be charged against the user's risk credit. This charge can be considered *the price paid for "purchasing" exceptional access to information and the necessary risk mitigation measures.* Periodically, the user's *return on investment* (ROI) will be evaluated; the return is the evaluation of the results delivered by the user, and the investment is the amount of risk charged. Greater reward will be given to those users with higher ROI. This process could be part of performance evaluations that an organization anyway conducts for its employees. A user's line of risk credit could be adjusted based on his/her ROI. The total risk for the organization is always below the sum of all lines of risk credit. Also, each "purchase" will be logged so the users' behaviors can be reviewed and the overall security policy, including boundaries of risk bands, risk mitigation measures, lines of risk credit, and users' MLS labels can be regularly fine-tuned to be more aligned with the actual needs. The lines of credit also provide a mean for users to tide over minor conflicts between their needs and the current policy in real–time, i.e., provides flexibility in the short term whereas the fine–tuning process which is to be done off–line adjusts the policy for long term trends.

Another system which extends the credit card approach above would be to create a *market-based mechanism* for users to "purchase risk" using a pseudo currency. There will be a finite number of risk units in the market based on the cap on risk that the organization is willing to accept. As before, exceptional accesses will need to be paid for by the users based on the difference between the risk of access and the soft boundary in risk units. Each user may be allocated some amount of risk units and pseudo currency initially to get her started, but there would be a market for users to buy and sell risk units for pseudo currency. To motivate prudent risk taking in such a market setting, a user's contribution to the organization will be evaluated periodically and a score will be given in an amount of pseudo–currency. It is important that pseudo-currency should have direct value to a user, possibly by linking it directly to actual monetary benefit. This way, a user who has knowledge and reason to believe that a particular risky access has a disproportionate chance of yielding benefit compared to current market rate for risk, would be motivated to purchase risk units from the market and pursue the opportunity; whereas users who do not see any good opportunity to use their risk units would be encouraged to sell their risk units in the market to acquire pseudo–currency. This way, it would be possible to aggregate the collective knowledge of the users [Sun95, Smi82, BSB03, Hay45] to optimally allocate the risk towards maximizing the benefit to the organization.

The exact market mechanism to be used and how it should be run (e.g., time periods for risk unit distribution and evaluations etc) would be subjects of further research.

The JASON report [JPO04] also presents some ideas on market mechanism; it discusses the notion of an *access token* which grants access right to certain kinds of access. The report gives the following example:

> *1 token = risk associated with one–day, soft–copy–only access to one document by the average Secret–cleared individual.*

A token associated with a specific kind of access is assigned a value using some common denomination. This allows different tokens, and therefore different access rights, to be traded. So it is more like a barter system; and the report does not present an uniform way nor a mathematical model to quantify the risk associated with information access or to compute the value of a token.

4 Related Work

Research on risk in access control models, flexible access control models, and risk management in general have been done for many years. We highlight a few recent ones that are more related to our work. The JASON report [JPO04] discusses the importance for a risk–based access control system in which the risk is measurable. McDaniel [McD03] discussed how the context of an access control decision can affect the decision. Nissanke and Khayat [NK04] analyzed the risk associated with permissions assigned to a role in an RBAC system where the risk is assessed by an independent assessment process. None of the works presented a way to quantify risk. Dimmoc et al. [DBIM05] discussed a computational approach to estimate risk and uses the estimate to make optimal decisions. However, the subjects in their model are autonomous agents, not humans; and it seems that the model requires a prior knowledge of outcomes of all possible combinations of states and actions when a decision is being made and we doubt if such knowledge is obtainable in general.

5 Fuzzy MLS Model : an Experiment of QRAAC

In this section we will discuss the Fuzzy MLS model. We will first discuss the context within which the model is developed and then the model itself. It is very hard to reason about risk without a context or the system in which risk management is practiced. The context within which the Fuzzy MLS model is developed is the "Brokerage of the Future" scenario that is described in section 5.1. Such a brokerage needs to constantly access, analyze and protect a large amount of sensitive and privileged information. This is also the scenario for the development of IBM System S [IBM06] which is an exploratory, very high performance data analysis system designed and built to continuously analyze a huge amount of input data flow. The Fuzzy MLS

model is the security policy model used for controlling human access to information provided by System S. Section 6 will discuss how the model is implemented and applied in System S.

5.1 The "Brokerage of the Future" Scenario

Consider a (futuristic) brokerage that has set up an information processing system for monitoring and analyzing different data sources such as news reports, trading data as well as other data from non-public, sensitive sources. This system is available to its traders, fund managers and brokers through a query interface that produces discrete and streaming results to the user's inquiries. Examples of such inquiries could be "what is the short term pricing trend for security X". Each inquiry can dynamically result in a chain of analysis being performed within the system using public as well as sensitive data that the brokerage has purchased at great cost or whose usage it doesn't want other business rivals to know about, and some of the analysis components themselves could utilize the brokerage's internal secrets such as models of markets. Different inquiries could dynamically result in different chain of analysis processing being performed on the input data to produce the results. Such a dynamic composition of analysis to respond to different inquiries is possible using planning techniques from AI [RL05, RL06].

For this system, data centric access control model such as MLS [BL76, Den76] with provision for data downgrading [Sto75, SRS+02, STH85] is more appropriate than user centric models since results are generated from a dynamic combination of analysis algorithms and data sources and a fundamental requirement is that access restrictions on any result should be easy to determine. With this approach data sources are labeled with sensitivity levels commensurate with the monetary loss if the information is disclosed. For important stocks, categories are used to protect stock–specific sensitive sources and algorithms. E.g., the brokerage estimates production figures for company X (a packaged food supplier) using revenue estimates of X's packaging material suppliers which are available from an expensive market intelligence newsletter. The brokerage also has a custom mathematical model for X's stock. Both the data source (newsletter) and the model are protected by the category X. Brokers and traders specializing in the packaged foods industry are cleared for category X and can receive detailed reports about X that include packaging revenues and stock model parameters. Other users not cleared for category X only get limited trend prediction for the stock X by means of downgraders. Valuable sources that can predict multiple stocks are given their own categories and downgraders are used to indicate their influence on each important stock. E.g., a sensitive source may be the daily sales figures for different food items from a major grocery chain. It is assigned category Y and a downgrader can utilize only a part of this data to produce competitive analysis of company X with respect to its peers.

However, any traditional access control model is not suitable in this dynamic environment. When there is a market anomaly, the brokerage would be willing to accept more risk rather than suffer huge losses and would temporarily want to allow wider access to sensitive information. But, when times are good it would want to exercise tighter control on sensitive information to avoid the risk of disclosure. Also, with this setup many traders will have their needs unmet and no satisfactory way to meet them. Consider a hedge fund manager, from time to time the manager needs to make huge short term bets on particular stocks, but requires detailed information before making the bet. In the MLS model, either the hedge fund manager needs to be given access to all major stock categories, giving him unfettered access to most of the brokerage's secrets or given no categories which mean he he gets only sanitized information about stocks and that doesn't serve his needs. Fuzzy MLS can solve both problems by having adjustable hard and soft boundaries that can globally (or just for the traders) be adjusted by security officer based on business conditions. The hedge fund manager will be given *partial, i.e., fuzzy memberships* in major stock categories and a *risk budget* so that, as needed, he can use his budget to get detailed information about any particular stock but that access consumes his budget and gets audited. This way, the hedge fund manager is able to perform his job, but the company can control its risks with respect to how much information the fund manager gets over time and audit records of what information he has accessed.

5.2 Fuzzy MLS: Computing Risk

The rationale for the MLS model was essentially risk based [DoD97] but it suffers from a binary decision model based on risk avoidance [JPO04]. Fuzzy MLS utilizes and extends the underlying risk based rationale of MLS but changes the access model to be based on risk management. For a *human user's read access*, the risk is defined as the expected value of loss due to unauthorized disclosure:

$$risk \;=\; (value\ of\ information) \;\times\; (probability\ of\ unauthorized\ disclosure) \tag{2}$$

The "value" of information is defined to be the damage sustained if this information is disclosed in an unauthorized manner, where units of damage would be organization specific. Estimating value may appear difficult but any organization already practicing MLS is expected to assign sensitivity levels to information based on a rough estimate of its value as prescribed by the principles in [DoD97]. *Typically, sensitivity levels correspond to order of magnitude of loss* and thus *approximate "value" can be derived from a traditional sensitivity level by an exponential function.*

Determining the probability of unauthorized disclosure requires more work. A precise determination is generally impossible since that would require a precise prediction of future actions of the user. Instead, the Fuzzy MLS model strives to develop a way to assign such probabilities that is commensurate with common sense and intuition which largely comes from prior research done on the traditional MLS model. For example, the probability should be very high when a person without security clearance is given access to top secret information but relatively low if the access is given to a person with top secret clearance. The Bell–Lapadula MLS model [BL76] can be viewed as estimating such a probability P from two probabilities P_1 and P_2 and combining them.

$$P_1 \;=\; \begin{cases} 0 & \text{human subject clearance level} \geq \text{object sensitivity level} \\ 1 & \text{otherwise} \end{cases}$$

$$P_2 \;=\; \begin{cases} 0 & \text{human subject category set} \supseteq \text{object category set} \\ 1 & \text{otherwise} \end{cases}$$

$$P \;=\; P_1 + P_2 - P_1 P_2 \tag{3}$$

The Fuzzy MLS model also estimates P_1 and P_2 but they are no longer binary.

5.2.1 Computing P_1

We consider P_1 to be the probability that a human subject leaks the information by succumbing to *temptation*. For a human user, the temptation would be a function of the user's clearance level (sl) which indicates the user's trustworthiness and object sensitivity level (ol) which indicates the value of the object. Temptation should monotonically increase with respect to ol and monotonically decrease with respect to sl. MLS takes a binary view of temptation: no temptation when $ol \leq sl$ and full temptation otherwise. MLS also uses a step function to relate temptation to the probability of disclosure P_1, no disclosure when there is no temptation and disclosure with probability 1 when there is temptation. We take a more nuanced view that all accesses result in temptation which we quantify by a temptation index TI that varies over a scale, TI is then converted to P_1. There could be countless many ways to derive TI which is a function of sl and ol but we submit that any such function should have the following properties that are consistent with our intuition:

- The more sensitive an object is, the higher the temptation,
  $ol_1 > ol_2 \Rightarrow TI(sl, ol_1) > TI(sl, ol_2)$.
- The more trustworthy a subject is, the lower the temptation,
  $sl_1 > sl_2 \Rightarrow TI(sl_1, ol) < TI(sl_2, ol)$

- TI is always greater than 0. This implies our belief that no human subject is above temptation nor completely trustworthy.
- TI is biased toward more sensitive objects.
  - The more sensitive an object is, the faster TI increases as sl decreases, $ol_1 > ol_2 \Rightarrow 0 > \partial TI(sl, ol_2)/\partial sl > \partial TI(sl, ol_1)/\partial sl$
  - For a constant difference $(sl - ol)$, TI increases as ol increases, $TI(sl_1, ol_1) > TI(sl_2, ol_2)$ if $ol_1 > ol_2$ and $(sl_1 - ol_1) = (sl_2 - ol_2)$.

As an example formulation for TI, we choose formula 4 below since it is simple, analytic and has all the above properties and some other nice properties as well . Let a be a real number that is greater than 1 and m be a real number that is greater than the maximum allowed value of ol. We further assume that sl and ol are non–negative, then

$$TI(sl, ol) = (a^{-(sl-ol)})/(m - ol) \tag{4}$$

Here a^{ol} *corresponds to the estimate value of loss as explained in section 5.2*; and a^{sl} *corresponds to the trustworthiness of a human subject*[2] such as "John can be trusted with information worthy of at most $10M". In this formulation TI approaches infinity as ol approaches m; the intuition behind m is that the temptation for a human subject is considered to be too great if an object is as sensitive as m or more sensitive than m and such access control decisions should not be made by machines. Formula 4 can also be easily related to the Bell–LaPadula model based MLS policy since TI is greater than $1/(m - ol)$ if $sl < ol$, less than $1/(m - ol)$ if $sl > ol$ and equal to $1/(m - ol)$ if $sl = ol$. Thus, with this formula we have that the the Bell–LaPadula model is violated iff TI is greater than $1/(m - ol)$.

P_1 should monotonically increase with TI. While there could be many different ways to relate TI to P_1, we choose a *sigmoid* function [Jyh97] in order to closely parallel the MLS step function approach. P_1 is defined as

$$P_1 = \frac{1}{1 + exp(\ (-k) \times (TI - mid)\ )} \tag{5}$$

where the parameter mid is the value of TI when P_1 is 0.5 and k determines the slope of the P_1 curve with regard to TI.

5.2.2 Computing P_2

Our intuition for P_2 comes from the probability of inadvertent disclosure. This is the probability that a human subject discloses the information *unintentionally*; this kind of "slip of tongue" is always possible once the information is in a human mind. When a human subject has a very strong, legitimate need for information in a category, the organization is more willing to accept this probability as the usual risk associated with conducting its business. When the subject only has marginal or no need, the organization is less willing to accept the probability. If a subject accesses an object belonging to only one category, P_2 is the difference between the probability of inadvertent disclosure and the probability which the organization is willing to accept for that subject; P_2 is zero if the difference is negative. If the object belongs to multiple categories, we make the *simplifying assumption that the object is a monolithic entity* and compute a difference for each category and use the maximum difference as P_2.

More research is needed to determine the probability of inadvertent disclosures for a category and an organization's willingness to accept such disclosures. We expect different categories to have different considerations for specifying the probability and willingness. For example, a category which exists purely to hide the existence of some information may have high probability of inadvertent disclosure and low willingness to accept disclosure by people without strong need to access information in the category. While we are currently experimenting with the following formulation to compute P_2, there could be other formulations or even explicit probability table listings. For a category c, a subject is given a *fuzzy membership* in $[0, 1]$ that indicates the

[2] See Appendix A for more details.

subject's need for information in the category; an object is also given a fuzzy membership that indicates the relevance of this object to the category. Thus the willingness decreases as the subject membership decreases and the object membership increases. The subject and object memberships can be used to compute a *willingness index* using formula 6 where $b > 1$, sm and om are subject and object memberships, and m_{max} is the maximum category membership.

$$wi_c(sm, om) \;=\; (b^{-(om-sm)})/(m_{max} - sm), \tag{6}$$

Formula 6 is similar to 4 but the bias is on the subject membership so the willingness decreases rapidly as the subject membership decreases. This index can be used in place of TI in formula 5 to compute w_c =*willingness to accept for* c, which is a number in $[0, 1]$.

$$w_c \;=\; \frac{1}{1 \;+\; exp(\;(-k') \times (wc_i - mid')\;)} \tag{7}$$

If P_c denotes the probability of inadvertent disclosure for category c,

$$P_2 \;=\; Maximum\{\; P_c(1 - w_c) \mid c \; is \; a \; category \;\} \tag{8}$$

5.2.3 Computing Risk

For a given subject and an object, the value of the object is computed from its sensitivity level, the probabilities P_1 and P_2 are computed as above, the probability of disclosure is computed using formula 3 and finally the risk is computed using formula 2.

5.3 Relationship to the Bell–LaPadula Model

As discussed in section 5.2, the Fuzzy MLS model quantifies the risk associated with a human subject's read access to information. Thus, the Fuzzy MLS model addresses the concern of the *simple security property* of the Bell–LaPadula model; this property states that a subject can not *read up* and is meant to prevent unauthorized disclosure of information to human subjects. In Fuzzy MLS a *read–up* shows up as higher risk that is to be managed.

The $*$*–property* of the Bell–LaPadula Model states that a non–human subject can not *write down* and is meant to prevent unauthorized disclosure of information through rogue or buggy programs. Such a program can mis–classify a piece of information and therefore give access to this information to a human subject who would not have the access otherwise. We note that in System S programs are not governed by Fuzzy MLS but by a lattice based access control system which does not allow read–up by any programs and also enforces the $*$*–property*, more details are provided in section 6. Any read-up by a human user is first authorized through Fuzzy MLS and then enforced by trusted components using a *escalated security label* deatiled in section 6.3. This escalated label is designed to prevent any write–down by a program.

6 Applying Fuzzy MLS in System S

In this section we will describe how the Fuzzy MLS model is implemented and applied in IBM System S [IBM06]. The scenario under which System S is developed is discussed in section 5.1. System S is an exploratory, grand challenge, prototype system being developed by IBM Research to support highly dynamic applications that extract information and knowledge by analyzing enormous volumes of relatively unimportant data. System S is designed to react quickly to events, changing requirements and priorities, handle orders of magnitude more data than existing systems, handle rapidly changing data formats and types and to constantly prioritize and adjust ongoing analysis since the amount of data and work will always exceed available computing resources. We will first give a brief introduction to System S and then describe how the Fuzzy MLS model is implemented and applied.

6.1 Introduction to IBM System S

As mentioned earlier, the purpose of System S is to process continuous flow of huge amounts of data. The explicit assumptions that went into the design of System S are that

- there is no restriction on the types of input data nor on the types of analysis done on the data,
- the actual analysises done are not pre–specified but constructed and instantiated during run time according to *inquiries* submitted by users.

Therefore rather than a system doing specific types of analysis on specific types of data, the architecture of System S is an extensible framework such that

- new data types and analyzing capabilities can be added dynamically,
- the job that answers an inquiry can be planned and instantiated dynamically.

The main System S abstractions for analyzing data are *streams* and *processing elements* (*PE*s). A stream is an *uni–directional flow of data objects* and can have *only one writer* but multiple readers. A PE is a program that can be instantiated to analyze input stream(s) to produce output stream(s). A stream is cataloged by System S upon its creation; the stream's catalog record includes the stream's name, producer, type information of data in the stream, *security labels* (described later) and other meta data. A PE is cataloged by System S when it is registered with System S; besides the PE's name, producer and other meta data, the catalog record contains specifications of the PE's *input* and *output ports* which are ingress and egress points for input and output streams. A port specification is basically a logical predicate on stream meta data that specifies what kinds of streams can go into or come out from the port.

When a user submits an inquiry, such as "What is the one month trend for stock X?", the *Planner* [RL06, RL05], using AI and other technologies, matches the inquiry with available streams and PEs in the catalogs to produce *plans*. A plan is a graph whose edges are streams and whose vertices are PEs; the input streams to the plan have no originating vertices in the graph and the final output streams have no terminating vertices. A plan can be instantiated as a *job* that runs to produce output to satisfy the inquiry. For an inquiry, the Planner is capable of producing multiple plans with *security* and other *constraints*; in particular, *the specifications for input ports of PEs in a plan are augmented with constraints on the security labels of streams which can flow into these ports*; output streams are also assigned security labels. Each such plan is annotated with a score of output quality, resource usages and other attributes, the plan and its annotation is presented to the user as an option. The user chooses one plan and submits the plan through the Planner to the *Scheduler* [WBF+]. At the Scheduler's discretion, the chosen plan may be instantiated as a running job. It is the Scheduler's duty to constantly monitor resource usages and re–provision the resources of all jobs.

Before describing how Fuzzy MLS is implemented in System S, we need to give a brief introduction to System S architecture. A simplified view of System S architecture has 3 layers as shown in Figure 2. The inner circle is the *stream processing core* (*SPC*) [DBH+06, AJS+06]; the outer layer is the interface to the outside world; the middle layer links the core to the outside world and offers some protection to the core from the outside in a way that is similar to the DMZ in a firewall setting [CB94]. The outer layer includes user interfaces for users to login and create a session, submit inquiries and view the results of inquiries (outputs of jobs); it also includes ingress points for external data sources and some pre–processing facilities for these external data. The middle layer includes user login/authentication servers, the Planner and a *result repository* to hold the results of inquiries; it also includes *data sanitizers* to check for the sanity (format correctness, . . .) of the external data and packages these data into streams which are fed to the SPC as input streams. The data sanitizers assign security labels to these input streams. Each login session is assigned a security label that is determined from a security label that is part of the user's login request and some other security constraints. The inner layer, the SPC, includes the Scheduler and a set of machines upon which jobs/PEs can be instantiated. The inner circle also has a *Data Graph Manager* (*DGM*) which connects an instantiated PE's input ports to streams by matching the ports' specifications to streams. These input streams may

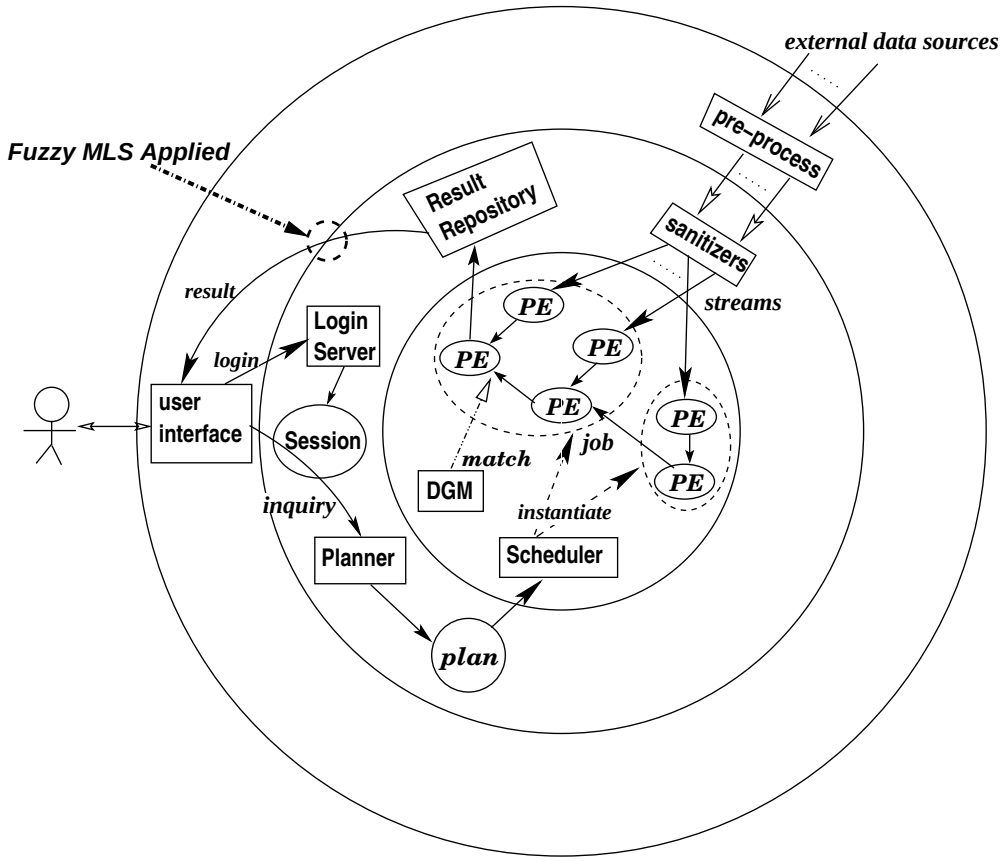

Figure 2: IBM System S Architecture

be constructed from external data sources or it may be output streams from other instantiated PEs.

The basic premise for information security in System S is that *the system owns and has total authority on the information (streams) and the PEs*. Since the objects (streams) and the subjects (PEs) are all owned by the system, there is *no need for Discretionary Access Control (DAC) within the system*. We chose to use a *Multi–Level Secure model for Mandatory Access Control (MAC) within System S*; *Fuzzy MLS is applied along the edge of the middle layer to quantify and control the amount of risk the system takes when disclosing sensitive outputs to users*. The System S MLS model is a lattice–based model that uses the Caernarvon model [SRS+02] for both data secrecy and data integrity. The Caernarvon model modifies both the Bell–LaPadula [BL76] secrecy model and the Biba [Bib77] integrity model to make secrecy downgrading and integrity upgrading an integral part of the model. The rules of the Biba model are changed to distinguish between reading low integrity data and running low integrity programs, and an integrity upgrading or a secrecy downgrading program (collectively termed a *guard application*) is assigned a specific range of security labels over which it may operate. A program's integrity level in the model should be assumed to be low unless it is formally evaluated [SRS+00, ISO98] and the integrity of a guard application and the range of security labels over which it may operate should be certified by an appropriate authority and digitally signed and cryptographically bound to the actual certified binary code module.

The System S MLS model further augments the Caernarvon model by introducing *fuzzy category membership*. A category set is a collection of fuzzy category memberships of all categories; and the *dominance* relationship between two Bell–Lapadula labels is augmented by the *superset* relationship between fuzzy sets [Jyh97]: a category set A is a superset of a category set B iff *for every category* c, $c's$ *membership in* $A \geq c's$ *membership in* B. Each user session, subject or object is assigned a *security label which is a combination of the Caernarvon secrecy label and*

the Caernarvon integrity label. A user is assigned *a range of security labels* from which the user can choose a label to submit as part of a login request, this label and other security considerations such as the user's physical location will determine the label for the session. *Security constraints are expressed as upper limits and lower limits on security labels*. Each input port of an instantiated PE has an upper limit on the security labels of its input streams. The label on an output stream of an instantiated PE is the *least upper bound* (*LUB*) of the PE's input limits. If the PE is a guard application, its range of input labels and its output label are part of the PE specification. An instantiated PE's output label is also the security label of the PE. *All security labels are assigned and maintained by trusted components*. The data sanitizers, login server, Planner, Scheduler, DGM and guard applications are all trusted components. The machines inside the SPC are controlled by trusted *secure hypervisors*. A secure hypervisor divides the machine into several *virtual machine partitions* within which PEs are running. A partition is assigned a security label and PEs with the same security labels as that of the partition can be placed in that partition. Therefore the labels of input streams to PEs in a partition are capped by the partition's label. All the input and output streams of PEs are mediated by the secure hypervisor and therefore the security labels of these streams are all maintained by the secure hypervisor. The use of secure hypervisor allows us to use commercial off–the–shelf, untrusted OSs and applications within virtual machine partitions but still maintain the correctness of the security labels. The initial prototypes of System S use existing hypervisors [SRJ+05, GBG+05]. These hypervisors do not actually support either the Bell–LaPadula or the Biba models and are not designed to high–assurance standards. However in an actual deployment, the use of a high–assurance hypervisor that supports the Caernarvon model would be essential. Fuzzy MLS itself could be an attractive target for a sophisticated penetrator, and only formally evaluated and certified high-assurance systems can resist such sophisticated attacks. Examples of such high-assurance hypervisors include KVM/370 [GLP+79] or the DEC A1-Secure VMM for the VAX [KZB+91]. KVM/370 supported the Bell–LaPadula model, and the DEC A1-Secure VMM supported both Bell–LaPadula and Biba models. A more in-depth discussion of the requirements for hypervisors that support MLS can be found here: [Kar05].

It should be noted that there is *no read–ups or write–downs by untrusted PEs* in System S; but by using Fuzzy MLS , System S may allow a human user read access to some sensitive outputs which are not allowed by the Caernarvon model, the details of which are explained in section 6.2.

6.2 Fuzzy MLS in IBM System S

For an organization running an instance of System S, the quantified risk can be treated as *a countable resource of some limited amount*. The limit is determined by the organization's *risk tolerance*. Using this idea, System S implements Fuzzy MLS in the following way, where the Planner acts as the decision maker and the DGM acts as the decision enforcer:

- The organization determines the maximum amount of risk its System S will take with respect to information disclosure, represented as an amount of quantified risk. Let's call this amount ORG_{CAP}.
- ORG_{CAP} is divided among the organization's employees who are the users of System S. For an employee U, his/her share of ORG_{CAP} is U_{CAP} which is the employee's risk budget. How to divide ORG_{CAP} among the employees remains on-going research and is discussed in section 3.
- U submits an inquiry to the Planner and indicates the maximum amount of risk that he/she is willing to take with the inquiry. Let's call this amount I_{CAP}. I_{CAP} must be less than or equal to U_{CAP}.
- The Planner uses I_{CAP} as one of its constraints to generate plans that could satisfy the inquiry. Each plan P is associated with an amount of quantified risk P_{risk} which is computed using Fuzzy MLS with the security labels of U and the plan's output. The Planner makes sure P_{risk} is less than or equal to I_{CAP} by assigning an upper limit on security label of the output from the plan. Once the planner knows the upper limit on

the label of the output, it tries to create a plan involving data sources, PEs and possibly downgraders so that the label of the output is dominated by this upper limit. Details on determining the upper limit are presented below.

- Each plan, together with its P_{risk} and other parameters, is presented to U. U can choose one plan and submit it for instantiation.
- If the chosen plan is instantiated, its P_{risk} is deducted from U_{CAP}. Thus the initial value of U_{CAP} limits the amount of risk U can take.
- After the plan is instantiated, the DGM matches streams to input ports of the plan's PEs and makes sure a stream's security label is dominated by that of the matching input port.

The current Planner [RL06, RL05] is capable of doing automatic planning using semantic knowledge. For an inquiry, it uses extensible knowledge bases and planning techniques to infer the types of output that may satisfy the inquiry and then uses this type information to find streams (as input) and PEs and downgraders that could be used to construct plans that produces the types of output. To make sure the risk associated with each plan is under I_{CAP}, the Planner iterates over each sensitivity level, starting from the lowest level, and determines:

- if U can access an output (an object) with this sensitivity level, and if so
- for each category, the maximum membership an output with this sensitivity level can have.

Since risk increases with output sensitivity levels, the iteration actually determines the maximum sensitivity level of output which U can access under I_{CAP}. This iteration process is described below:

For each sensitivity level, starting from the lowest level,

1. use the sensitivity level as the output sensitivity level (ol) and formula 2 to compute the probability P associated with I_{CAP} as $P = I_{CAP}/a^{ol}$. Where a^{ol} is the estimate value of loss as explained in sections 5.2 and 5.2.1. Stop if P is larger than 1.
2. use ol and the sensitivity level of U as sl, compute P_1 using formulas 4 and 5. Stop if P_1 is larger than P.
3. use formula 3 to compute the maximum allowed value for P_2 as $P_{2\_max} = (P-P_1)/(1-P_1)$.
4. referring to formula 8, for each category c, compute the output's maximum membership in c
   (a) compute the minimum value for *willingness to accept for* c : $w_{c\_min} = 1-P_{2\_max}/P_c$ if $P_{2\_max} < P_c$, otherwise set the maximum membership in c for the output to 1 and go to the next category.
   (b) use w_{c_min} and formula 7 to compute the minimum wi_c and then use formula 6 to compute the maximum membership in c for the output.

With the upper limits on output sensitivity level and category membership, the planner can generate multiple plans associated with different risk estimates, all capped by I_{CAP}. For example, for each output sensitivity level the planner can generate a plan under these upper limits.

6.3 Escalated Labels: Bridging Fuzzy MLS and MLS

If Fuzzy MLS allows a user to see an output which the user would not be allowed to see under the Caernarvon model; we say the user is temporarily *escalated*[3] to see the output. A guard application would make a *copy* of the output and this copy is assigned an *escalation label* to mark this exceptional access, why this copy is needed is explained later. An escalation label is defined as:

$$escalation\ label\ C\ =\ <fromL,\ toL,\ C_{tag}> \tag{9}$$

[3]This idea is from an author and a colleague; we omit the colleague's name for now due to anonymous submission.

Where $fromL$ is the output's true security label used in risk calculation, toL is the user's security label and C_{tag} is an unique ID for this exceptional access. C means that the copy is moved to the user's label so that the user can see the copy. The user would have the label C added to his/her range of security labels and could then login with the label C to see the copy. C_{tag} makes C unique so that the user cannot see another piece of information with a $< fromL,\ toL,\ A_{tag} >$ label without going through Fuzzy MLS . We extend the *dominance* relationship to escalation labels in a way that makes sure an escalation label does not create any additional allowed information flow in an MLS lattice. The extended relationship is defined as:

For an ordinary label X:

- X dominates $C \iff X$ dominates $formL$ and X dominates toL
- C dominates $X \iff toL$ dominates X

For another escalation label $D =< f_D,\ t_D,\ D_{tag} >$

- C dominates $D \iff C = D$ or (toL dominates f_D and toL dominates t_D)

With this definition, it can be shown that for another ordinary label Y:

- Y dominates C and C dominates $X \Rightarrow Y$ dominates toL and toL dominates $X \Rightarrow$ Y dominates X
- For a series of escalation labels $C_i,\ i = 1, \ldots, n$ such that Y dominates C_n, C_n dominates C_{n-1}, ..., C_1 dominates $X \Rightarrow Y$ dominates X since C_i dominates X if C_i dominates C_{i-1} and C_{i-1} dominates X.

The extended dominance relationship implies that a user with a security label dominating toL but not $fromL$ cannot see the copy without going through Fuzzy MLS , and a user with a security label dominating $fromL$ may not be allowed to see the copy but he/she can see the original output.

7 On–Going and Future Research

Besides the "risk market" discussed in section 3, our experience with the Fuzzy MLS prototype as part of System S (see section 6) shows some apparent advantages of Fuzzy MLS that may help address issues that arise in current MLS systems. These will be the subject of further research, we briefly discuss these ideas and some other topics.

- *Label Uncertainty*: In an MLS system labels are assumed to be correct. Also most MLS systems include the notion of *perfect* secrecy downgraders that sanitize data [Sto75, STH85, SRS+02]. This situation makes data difficult to share because human labelers and downgraders err on side of higher secrecy. If labels were uncertain or probabilistic to account for the uncertainty in ascertaining the right secrecy level, then the situation of overclassification could be addressed. But traditional MLS model cannot make access decisions with labels having uncertainty. However this is not a problem with Fuzzy MLS since it can still compute the risk associated with the access and make the decision based on risk.
- *Loss variance based access decisions*: Fuzzy MLS can be easily extended so that both the expected loss and the variance of loss can be used to make access decisions; users may have variance in their trustworthiness and data may have variance with respect to their secrecy and these can be combined to compute risk and variance of loss for making access decisions.
- *Aggregation Problem*: This problem has not been satisfactorily resolved in MLS systems, even in the simplest form, where a sequence of allowed accesses to less sensitive data results in a collection of data that is more sensitive than the individual items. With Fuzzy MLS, the aggregation problem gets exposed, as each user access to data incurs a risk and multiple accesses should accumulate risk. We are exploring ways in which label uncertainty can be used to address the aggregation problem. If individual data items have

label X but collectively have higher secrecy label Y, one idea is to assign the individual data item an uncertain label which indicates that it has a small probability of having the label Y. Repeated access to such objects would then incur an aggregate risk comparable to an access to Y.

- *Risk Modulating Factors*: Many factors other than security labels, such as usages of risk mitigation measures, security of the physical environments, properties of information delivery channels (hard/soft copies, use of cryptography, ...) can affect risk estimates. Taking these risk modulation factors into account will result in a more holistic and realistic model for dynamic environments found in mobile settings.
- *Fine–Tuning Parameters*: Parameters used in computing quantified risk estimates have to be fine tuned to account for changes in environments, needs, better understanding of risk contributing factors, etc. Better risk estimating can be learned over time if proper models can be developed.

8 Conclusion

We have presented the general idea of quantified risk–adaptive access control as a possible replacement for traditional access control models to achieve flexible access control that can meet present and future needs while constantly trading off risk against benefit of granting access and capping the total risk taken. We have also presented Fuzzy MLS as a realization of the idea in a information flow context and the feasibility of Fuzzy MLS is demonstrated by our prototype. Experiences with traditional access control models [JPO04] also shows risk–adaptive access control is the way to go. In particular, the ability to compute quantified risk estimates allows risk to be treated as countable resources and opens the door to promising research such as using market mechanism and economy principles to achieve optimal risk vs. benefit tradeoff. Making access control decisions based on quantified risk estimates also makes it possible to solve lingering MLS problems such as label uncertainty and aggregation.

9 Acknowledgment

We thank Shai Halevi, Trent Jaeger, Ronald Perez, Michael Steiner, Douglas L. Schales, Jeff Kravitz and Josyula R. Rao for valuable suggestions.

References

[AJS+06] Lisa Amini, Nevendu Jain, Anshul Sehgal, Jeremy Silber, and Olivier Verscheure. Adaptive control of extreme-scale stream processing systems. In *the 26th International Conference on Distributed Computing Systems*. IEEE, July 2006. 11

[Bib77] K.J. Biba. Integrity Considerations for Secure Somputer Systems. Technical Report ESD–TR–76–732, The MITRE Corporation, Bedford, MA. HQ Electronic Systems Division, Hanscom AFB, MA., April 1977. 12

[BL76] David E. Bell and Leonard J. LaPadula. Computer Security Model: Unified Exposition and Multics Interpretation. Technical Report ESD–TR–75–306, The MITRE Corporation, Bedford, MA. HQ Electronic Systems Division, Hanscom AFB, MA, March 1976. http://csrc.nist.gov/publications/history/bell76.pdf. 7, 8, 12

[BSB03] Andrew Byde, Mathias Sallé, and Claudio Bartolini. Market–Based Resource Allocation for Utility Data Centers. Technical Report HPL–2003–188, HP Laboratories Bristol, September 2003. 6

[CB94] William R. Cheswick and Steven M. Bellovin. *Firewalls and Internet Security Repelling the Wily Hacker*. Addison Wesley, 1994. 11

[CC03] Suresh N. Chari and Pau-Chen Cheng. BlueBox: A Policy–Driven, Host–Based Intrusion Detection System. *ACM Transactions on Information and System Security*, 6(2), May 2003. 3, 20

[DBH+06] Fred Douglis, Michael Branson, Kirsten Hildrum, Bin Rong, and Fan Ye. Multi-site cooperative data stream analysis. *ACM SIGOPS Operating Systems Review*, 40(3), July 2006. 11

[DBIM05] Nathan Dimmock, Jean Bacon, David Ingram, and Ken Moody. Risk Models for Trust–Based Access Control (TBAC). In *the Third Annual Conference on Trust Management (iTrust 2005)*. Springer–Verlag, May 2005. 6

[Den76] Dorothy E. Denning. A lattice model of secure information flow. *Communications of the ACM*, 19(5):236 – 243, May 1976. ISSN 0001–0782. 7

[DoD97] INFORMATION SECURITY PROGRAM, DOD 5200.1–R, US Department of Defense, January 1997. http://www.fas.org/irp/doddir/dod/5200-1r/. 4, 8

[GBG+05] John L. Griffin, Stefan Berger, Kenneth A. Goldman, Trent R. Jaeger, Ronald Perez, David R. Safford, Reiner Sailer, Enriquillo Valdez, Leendert P. Van Doorn, and Xiaolan Zhang. Secure Foundations for Mission-Critical Computing. In *CIIP – 2nd Japan/U.S. Workshop on Critical Information Infrastructure Protection*, May 2005. 13

[GLP+79] B.D. Gold, R.R. Linde, R.J. Peeler, M. Schaefer, J.F. Scheid, and P.D. Ward. A Security Retrofit of VM/370. In *AFIPS Conference Proceedings*, volume 48, pages 335–344. National Computer Conference, 1979. Montvale, NJ: AFIPS Press. 13

[Hay45] F. A. Hayek. The Use of Knowledge in Society. *American Economic Review*, 30:519–530, 1945. 6

[IBM06] Security for IBM System S, 2006. http://domino.research.ibm.com/comm/research_projects.nsf/pages/system_s_security.index.html. 6, 10

[ISO98] ISO/IEC 15408. Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Models, May 1998. http://www.radium.ncsc.mil/tpep/library/ccitse/index.html. 12

[JPO04] MITRE Corporation Jason Prograrm Office. HORIZONTAL INTEGRATION: Broader Access Models for Realizing Information Dominance, JSR–04–132, December 2004. http://www.fas.org/irp/agency/dod/jason/classpol.pdf. 1, 3, 6, 8, 16

[Jyh97] Jyh–Shing Roger Jang and Chuen-Tsai Sun and Eiji Mizutani. *Neuro–Fuzzy AND Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, 1997. ISBN 0–13–261066–3. 1, 9, 12

[Kar05] P.A Karger. Multi–Level Security Requirements for Hypervisors. In *21st Annual Computer Security Applications Conference,* Tucson, AZ, pages 240–248. IEEE Computer Society, 2005. http://www.acsa-admin.org/2005/papers/154.pdf. 13

[KZB+91] P.A. Karger, M.E. Zurko, D.W. Bonin, A.H. Mason, and C.E. Kahn. A Retrospective on the VAX VMM Security Kernel. *IEEE Transactions on Software Engineering*, 17(11):1147–1165, 1991. 13

[McD03] Patrick McDaniel. On Context in Autorization Policy. In *SACMAT*, Como, Italy, June 2003. 6

[NK04] Nimal Nissanke and Etienne J. Khayat. Risk Based Security Analysis of Permissions in RBAC. In *2nd International Workshop on Security in Information Systems*, Porto, Portugal, April 2004. 6

[RL05] Anton Riabov and Zhen Liu. Planning for Stream Processing Systems. In *The Twentieth National Conference on Artificial Intelligence*, AAAI 2005. 7, 11, 14

[RL06] Anton Riabov and Zhen Liu. Scalable Planning for Distributed Stream Processing Systems. In *The International Conference on Automated Planning and Scheduling (ICAPS)*, 2006. 7, 11, 14

[Smi82] Vernon L. Smith. Markets as Economizers of Information: Experimental Examination of the "Hayek Hypothesis". *Economic Inquiry*, 20(2):165–175, 1982. 6

[SRJ+05] Reiner Sailer, Trent R, Enriquillo Valdez Jaeger, Ronald Perez, Stefan Berger, John L. Griffin, Leendert P. Van Doorn, and Ramon Caceres. Building a MAC-based Security Architecture for the Xen Opensource Hypervisor. In *21st Annual Computer Security Applications Conference (ACSAC)*, September 2005. 13

[SRS+00] G. Schellhorn, W. Reif, A. Schairer, P. Karger, V. Austel, and D. Toll. Verification of a Formal Security Model for Multiapplicative Smart Cards. In *6th European Symposium on Research in Computer Security (ESORICS 2000), Toulouse, France*, pages 17–36. Springer–Verlag Lecture Notes in Computer Science Vol. 1895, October 2000. 12

[SRS+02] Gerhard Schellhorn, Wolfgang Reif, Axel Schairer, Paul Karger, Vernon Austel, and David Toll. Verified formal security model for multiapplicative smart cards. *Journal of Computer Security*, 10(4):339–367, 2002. 7, 12, 15

[STH85] R. R. Schell, T. F. Tao, and M. Heckman. Designing the GEMSOS security kernel for security and performance. In *the 8th National Computer Security Conference*, pages 108–119. DoD Computer Security Center and National Bureau of Standards, 1985. 7, 15

[Sto75] D.F. Stork. Downgrading in a Secure Multilevel Computer System: The Formulary Concept. Technical Report MTR 2924, ESD–TR–75–62, The MITRE Corporation: Bedford, MA. HQ Electronic Systems Division, Hanscom AFB, MA, May 1975. 7, 15

[Sun95] Shyam Sunder. Experimental Asset Markets: A Survey. In John H. Kagel and Alvin E. Roth, editors, *the Handbook of Experimental Economics*, pages 445–500. Princeton University Press, 1995. 6

[WBF+] Joel L. Wolf, Nikhil Bansal, Lisa Fleischer, Kirsten Hildrum, and Deepak Rajan. *SODA: A Scheduler for Large Stream-Based Computing Systems.* Research Report (to appear). 11

A Temptation Index Formula: Motivation and Example

In Section 5.2.1, we proposed formula 4 to compute the Temptation Index TI.

$$TI(sl, ol) \;=\; (a^{-(sl-ol)})/(m - ol)$$

The basic motivation for this formula follows from experience with MLS. We expect the value of an object to increase exponentially with the sensitivity level. Also, the way the user clearance process works today, subject labeled to a certain clearance level are allowed accesses to objects up to that level. This means that a measure of subject trustworthiness also increases exponentially with the subject level and at a rate which is probably commensurate with the rate at which object value increases. So a basic formula TI' where

$$TI'(sl, ol) \;=\; a^{-(sl-ol)}$$

captures the basis intuition in MLS that the temptation from an exponentially rising object value can be balanced by exponentially rising subject clearance level. TI' has the property that when its above 1 access is denied. But it does not conform to our intuition that given equal differences between subject and object levels, temptation should increase with higher object level. Therefore we tweaked TI' to create TI which is biased towards higher value objects.

Table 1 shows a table of temptation indices and Table 2 shows the corresponding probabilities. The main point to make here is that *temptation are indices usually fairly large or fairly low except the cases where the subject and object levels are close. This is the place where calculated risk taking should be allowed.*

$ol \backslash sl$	1	2	3	4	5
1	$1.000e-01$	$1.000e-02$	$1.000e-03$	$1.000e-04$	$1.000e-05$
2	$1.111e+00$	$1.111e-01$	$1.111e-02$	$1.111e-03$	$1.111e-04$
3	$1.250e+01$	$1.250e+00$	$1.250e-01$	$1.250e-02$	$1.250e-03$
4	$1.429e+02$	$1.429e+01$	$1.429e+00$	$1.429e-01$	$1.429e-02$
5	$1.667e+03$	$1.667e+02$	$1.667e+01$	$1.667e+00$	$1.667e-01$
6	$2.000e+04$	$2.000e+03$	$2.000e+02$	$2.000e+01$	$2.000e+00$
7	$2.500e+05$	$2.500e+04$	$2.500e+03$	$2.500e+02$	$2.500e+01$
8	$3.333e+06$	$3.333e+05$	$3.333e+04$	$3.333e+03$	$3.333e+02$
9	$5.000e+07$	$5.000e+06$	$5.000e+05$	$5.000e+04$	$5.000e+03$
10	$1.000e+09$	$1.000e+08$	$1.000e+07$	$1.000e+06$	$1.000e+05$

$ol \backslash sl$	6	7	8	9	10
ol	6	7	8	9	10
1	$1.000e-06$	$1.000e-07$	$1.000e-08$	$1.000e-09$	$1.000e-10$
2	$1.111e-05$	$1.111e-06$	$1.111e-07$	$1.111e-08$	$1.111e-09$
3	$1.250e-04$	$1.250e-05$	$1.250e-06$	$1.250e-07$	$1.250e-08$
4	$1.429e-03$	$1.429e-04$	$1.429e-05$	$1.429e-06$	$1.429e-07$
5	$1.667e-02$	$1.667e-03$	$1.667e-04$	$1.667e-05$	$1.667e-06$
6	$2.000e-01$	$2.000e-02$	$2.000e-03$	$2.000e-04$	$2.000e-05$
7	$2.500e+00$	$2.500e-01$	$2.500e-02$	$2.500e-03$	$2.500e-04$
8	$3.333e+01$	$3.333e+00$	$3.333e-01$	$3.333e-02$	$3.333e-03$
9	$5.000e+02$	$5.000e+01$	$5.000e+00$	$5.000e-01$	$5.000e-02$
10	$1.000e+04$	$1.000e+03$	$1.000e+02$	$1.000e+01$	$1.000e+00$

Table 1: TI values, $m = 11.0,\ a = 10.0$

$ol \backslash sl$	1	2	3	4	5
1	$5.215e-02$	$4.788e-02$	$4.747e-02$	$4.743e-02$	$4.743e-02$
2	$1.314e-01$	$5.271e-02$	$4.793e-02$	$4.748e-02$	$4.743e-02$
3	$9.999e-01$	$1.480e-01$	$5.340e-02$	$4.799e-02$	$4.748e-02$
4	$1.000e+00$	$1.000e-00$	$1.720e-01$	$5.431e-02$	$4.808e-02$
5	$1.000e+00$	$1.000e+00$	$1.000e-00$	$2.086e-01$	$5.555e-02$
6	$1.000e+00$	$1.000e+00$	$1.000e+00$	$1.000e-00$	$2.689e-01$
7	$1.000e+00$	$1.000e+00$	$1.000e+00$	$1.000e+00$	$1.000e-00$
8	$1.000e+00$	$1.000e+00$	$1.000e+00$	$1.000e+00$	$1.000e+00$
9	$1.000e+00$	$1.000e+00$	$1.000e+00$	$1.000e+00$	$1.000e+00$
10	$1.000e+00$	$1.000e+00$	$1.000e+00$	$1.000e+00$	$1.000e+00$

$ol \backslash sl$	6	7	8	9	10
1	$4.743e-02$	$4.743e-02$	$4.743e-02$	$4.743e-02$	$4.743e-02$
2	$4.743e-02$	$4.743e-02$	$4.743e-02$	$4.743e-02$	$4.743e-02$
3	$4.743e-02$	$4.743e-02$	$4.743e-02$	$4.743e-02$	$4.743e-02$
4	$4.749e-02$	$4.743e-02$	$4.743e-02$	$4.743e-02$	$4.743e-02$
5	$4.818e-02$	$4.750e-02$	$4.743e-02$	$4.743e-02$	$4.743e-02$
6	$5.732e-02$	$4.834e-02$	$4.752e-02$	$4.743e-02$	$4.743e-02$
7	$3.775e-01$	$6.009e-02$	$4.857e-02$	$4.754e-02$	$4.744e-02$
8	$1.000e-00$	$5.826e-01$	$6.497e-02$	$4.895e-02$	$4.758e-02$
9	$1.000e+00$	$1.000e+00$	$8.808e-01$	$7.586e-02$	$4.974e-02$
10	$1.000e+00$	$1.000e+00$	$1.000e+00$	$9.991e-01$	$1.192e-01$

Table 2: Probability for TI values in Table 1, $k = 1.0$, $mid = 3.0$

B Risk Mitigation Measures

In Section 2.2, we introduced the notion of risk mitigation measures; in section 3 we introduced the idea of using risk units as currency for purchasing risk mitigation measures for risky access. We describe possible risk mitigation measures in more detail here. Since a subject cannot be made more trustworthy instantly, risk mitigation measures are geared towards making the subject less likely to disclose information. Such measures usually fall into the following categories : *deterrence*, *prevention* and *limiting damage* which are discussed below.

- **Deterrence**: provide (strong) disincentives for wrong doings. For example, detailed auditing and mandatory computer or human review may be used to ensure that risky accesses are made for the right reasons. This could also set the stage for administrative or legal actions.
- **Prevention**: To prevent a user process (not the user) from inadvertently disclosing information, one can insist on that the user process runs in a special environment which has extra physical and/or logical security. For example, exceptional accesses may require the process to run within a specific secure location or on a specific trustworthy system, or the process could be sandboxed [CC03].
- **Limiting Damage**: to assume that bad things will happen and take precaution measures to limit the potential damage. Examples are limiting the output rate of information flow to a user/user process, reduced scheduling priority, etc. Another measure is to further restrict the user's future access to resources based on the already granted accesses.