Towards Risk-aware Access Control Framework for Healthcare Information Sharing

Mohamed Abomhara, Geir M. Køien, Vladimir A. Oleshchuk and Mohamed Hamid
Department of Information and Communication Technology, University of Agder, Grimstad, Norway

Keywords: Access Control, Healthcare Information Sharing, Electronic Health Records, Security, Privacy, Risk Assessment, Risk Management.

Abstract: Access control models play an important role in the response to insider threats such as misuse and unauthorized disclosure of the electronic health records (EHRs). In our previous work in the area of access control, we proposed a work-based access control (WBAC) model that strikes a balance between collaboration and safeguarding sensitive patient information. In this study, we propose a framework for risk assessment that extend the WBAC model by incorporating a risk assessment process, and the trust the system has on its users. Our framework determines the risk associated with access requests (user's trust level and requested object's security level) and weighting such risk against the risk appetite and risk threshold of situational conditions. Specifically, an access request will be permitted if the risk threshold outweighs the risk of granting access to information, otherwise it will be denied.

1 INTRODUCTION

Electronic health records (EHRs) are an essential information source and a communication channel to create a shared view of the patient among healthcare providers (healthcare professionals and healthcare organizations) (Chao, 2016; Reitz et al., 2012). However, since EHRs systems gather and enable health information access, secure control over information flow is a key aspect of EHRs where sensitive information is shared among a group of people within or across organizations (Shoniregun et al., 2010). Insider abuse or misused of health information may lead to a significant damage to patient's privacy and healthcare provider alike (Probst et al., 2010). It is, on the one hand, healthcare providers are authorized to access their patient health information during patient treatment and on the other hand, there is always a security risk associated with each access to health information which might lead to potential information leakage (Salim et al., 2011; Rostad et al., 2007).

Access control is the most popular approach for developing an active form of mitigating insider threats (Baracaldo and Joshi, 2013; Probst et al., 2010). Role-based access control (RBAC) (Ferraiolo et al., 2001) and attribute-based access control (ABAC) (Hu et al., 2014) are a popular models for access control and they widely employed in medical industry (Ferraiolo et al., 2001; Salim et al., 2011). These access control models are characterized by a considerable imbalance between security and efficiency, especially when applied in distributed environments (Salim et al., 2010). One of the main problem in these models often is having a difficulties to enforce the "need-to-know" principle (Ferraiolo et al., 2001) and "minimum necessary" standard to use and disclose patient's records for treatment (Agris, 2014). This is due to problems like, for one, medical records containing a wide range of information and it is infeasible for the policy author (e.g., the administrator) to foreseen what health information a healthcare provider may need in various situation (Salim et al., 2011). Second, healthcare providers cannot decide on what appropriate information is really necessary in a patient's treatment case. Thus, healthcare provider get unlimited access to patient health records which lead to unaccountable risk on patient health records.

In our previous works (Abomhara and Køien, 2016; Abomhara et al., 2017), an enhanced access control model was proposed to strike a balance between collaboration and safeguarding sensitive patient information. The enhanced model introduced the team role concept and modifying the user-role assignment model from a RBAC and ABAC. In our access control model, the level of fine-grained control of access to objects that can be authorized to healthcare

Abomhara, M., Køien, G., Oleshchuk, V. and Hamid, M.
Towards Risk-aware Access Control Framework for Healthcare Information Sharing.
DOI: 10.5220/0006608103120321
In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pages 312-321
ISBN: 978-989-758-282-0

providers is managed and controlled based on the job required to meet the "minimum necessary" standard. In this study, we propose a framework for risk assessment to estimate the risk that users may present to objects by computing the risk of each user access request and the object accessed by the user. Access decisions are made by determining the risk associated with access requests (user's trust level and requested object's security level) and weighting such risk against the risk appetite and risk threshold of situational conditions. Specifically, an access request will be permitted if the risk threshold outweighs the risk of granting access to information, otherwise it will be denied.

The remaining part of this study is organized as follows. Section 2 presents a related work. In section 3, we briefly describe the WBAC model. Section 4 introduces the risk aware framework. In section 5, we present a usage scenario and proof of concept of the proposed framework. A discussion, conclusions and future works are provided in section 6.

2 RELATED WORK

To determine whether user access risk levels are on par with user trust levels and object security levels, Sandhu, 1993 proposed lattice-based access control models, where a user is only allowed to access an object if the security level of the subject is higher than or equal to the security level of the object.

Cheng, et al., 2007 proposed a quantified risk-adaptive access control based on fuzzy multi-level security. They illustrated the concept of their approach by showing how the rationale of the Bell-LaPadula model (Bell and LaPadula, 1975) and Multi-Level Security (MLS) access control model could be used to develop a risk-adaptive access control model. Chari, et al., 2012 used a fuzzy logic risk inferencing system to determine permission sensitivity levels and user access risk levels, and infer the risk of a particular role by looking at the aggregate role sensitivity and aggregate user access risk level.

Ma, 2012 presented a formal approach to risk assessment for RBAC systems. The basic idea of this approach is assigning a security level to each user, calculating the security level for role and then calculate the risk value of role-user assignment relation. Bijon, et al., 2013 discussed the difference between traditional constraint-based risk mitigation and recent quantified risk-aware approaches in RBAC, and also proposed a framework for a quantified approach to risk-aware role-based access control.

Baracaldo and Joshi, 2013 proposed a framework that extends the RBAC model. Their framework adapts to suspicious changes in users' behavior by removing privileges when users' trust falls below a certain threshold. Moreover, Shaikh et al., 2011 proposed a dynamically user trust calculation model based on the past behavior of the users with particular objects. The past behavior is evaluated based on the history of reward and penalty points assigned to user after the completion of every transaction. In their model, the old and recent history (rewards and penalty points history) has an equal weight values (Shaikh et al., 2012) and the consequences of this is that the model may be enable to detect small changes in recent behavior of user in timely manner. In this case, if the rewords points increase then the user trust level increase and vice versa. This model was extended in (Shaikh et al., 2012) by exponentially weighted moving average approach. However, in the extended model, exponentially weighted average of past behavior was obtained recursively. Also, the authors did not enforce giving the recent events higher weight than the old events. In this case, we could assume that, the insider would behavior according to the rules to increase his/her rewards points and reaches to the trust level that will allow him/her to violate system rules.

Motivated by the shortcomings in the related work, in our risk-aware framework, we calculate the risk associated with an access request using the trust level based on object security level (section 4.2) and the past user behavior. The past behavior of users is dynamically calculated based on rewards and penalty points assigned to users. In our model, rewards and penalty points history are given a different weight values based on the oldness of their occurrence based on the forgetting factor which allows weight the recent events higher than old one (section 4.3).

3 WORK BASED ACCESS CONTROL MODEL (WBAC)

WBAC enforces a three-layer access control that applies RBAC, a secondary RBAC and ABAC. The secondary RBAC layer, with extra team roles extracted from the team work requirements, is added to manage the complexity of cooperative engagements in the healthcare domain (Abomhara and Køien, 2016). Role and team role are used in conjunction to deal with access control in dynamic collaborative environments. Policies related to collaboration and team work are encapsulated within this coordinating layer to ensure that the attribute layer is not overly burdened (Abomhara et al., 2017).

3.1 WBAC Core Components

The Core WBAC model includes the following components:

- USR is a set of users.
- OBJ is a set of objects. $OBJ = OBJ_A \cup OBJ_B$ where, OBJ_A is a set of *private* objects, OBJ_B is a set of *protected* objects and $OBJ_A \cap OBJ_B = \emptyset$.
- OPR is set of operations.
- PER is set of permissions; $PER \subseteq OBJ \times OPR$.
- R is a set of roles.
- TR is a set team roles. $TR = \{tr_t, tr_a, tr_m\}$ where tr_t is *thought* role, tr_a is *action* role, tr_m is *management* role (early described in (Abomhara and Køien, 2016)).
- T is a set of teams, where $T \subseteq 2^{USR}$.
- W is a set of collaborative works. $W = (t, obj, state)$, where, $t \subseteq T$, $obj \subseteq OBJ$, and $state \in \{Active, Inactive\}$.
- $USR\text{-}R\text{-}A \subseteq USR \times R$: User to role assignment relation.
- $USR\text{-}T\text{-}A \subseteq USR \times T$: User to team assignment relation.
- $TM\text{-}TR\text{-}A \subseteq USR \times T \times TR$: Team member to a team role assignment relation.
- $PER\text{-}R\text{-}A \subseteq PER \times R$: Permission to role assignment relation.
- $PER\text{-}TR\text{-}A \subseteq PER \times TR$: Permission to team role assignment relation.
- $T\text{-}W\text{-}A \subseteq T \times W$: Team to work assignment relation.
- PS: Represents a policy set which is based on one or more policy.
- P: Represents a policy which composed of one or more rules.
- *rule*: Represents a rule in policy.

Definition 1. *Access state (Γ) is a set of all WBAC model assignments which contains all the information necessary to make access control decisions.*

In order to specify the access control, we define the following functions:

- $authorized\text{-}usr_{role}(r) = \{usr \in USR | (usr, r) \in USR\text{-}R\text{-}A\}$: A set of all users assigned to role r.
- $team_{member}(t) = \{usr \in USR | (usr, t) \in USR\text{-}T\text{-}A\}$: A set of team members in team t.
- $authorized\text{-}usr_{tr}(usr, t) = \{tr \in TR | (usr, t, tr) \in TM\text{-}TR\text{-}A\}$: A set of team roles hold by user usr in team t.
- $team\text{-}role_{member}(tr) = \{usr \in USR | \exists t \in T : tr \in authorized\text{-}usr_{tr}(usr, t)\}$: A set of all users assigned to team roles in team t.
- $assigned\text{-}role_{per}(r) = \{per \in PER | (per, r) \in PER\text{-}R\text{-}A\}$: A set of permissions hold by role r.
- $assigned\text{-}team\text{-}role_{per}(tr) = \{per \in PER | (tr, per) \in PER\text{-}TR\text{-}A\}$: A set of permissions hold by team role tr.
- $assigned\text{-}team\text{-}work(w) = \{t \in T | (w, t) \in T\text{-}W\text{-}A\}$: A set of teams assigned to work w.
- $can\text{-}access(usr, per)$: User usr has a permission per.

Definition 2. *An access policy is a security statement of what is, and what is not allowed in the organization. Formally: $can\text{-}access(usr, per) \iff rule_1 \wedge \ldots \wedge rule_n$, where $usr \in USR$, $per \in PER$ and $per = (obj, opr)$.*

It can be interpreted as user (usr) can perform an operation (opr) on object (obj) if and only if all conditions ($rule_i$) hold.

Definition 3. *A rule is a fundamental component of an policy. A rule consists of a condition (ruleCon) and an effect (ruleEff) that can be either a permission or denial associated with the successful evaluation of the rule. Formally: $rule = (ruleId, ruleCon, target, ruleEff, CombiningAlg)$.*

A rule condition is a Boolean function over subject, object and operation that refines the applicability of the rule beyond the predicates implied by its target. The correct evaluation of a condition returns the effect of the rule (permit or deny), while incorrect evaluation results in an error (*Indeterminate*) or the discovery that the condition does not apply to the request (*Not Applicable*). $target = (usr, obj, opr)$ is a set of attributes and their values for matching the user, object, and operation, to check if the given rule and *CombiningAlg* is the policy combining algorithm (e.g., deny-overrides, permit-overrides and/or first-applicable algorithms (Li et al., 2009)).

Example 1. *Consider a policy with one rule ($rule_1$) to ensuring that the primary doctor has a clearance to read medical file. Formally: $rule = (Rule_1, \{isPrimaryDoctor\}, \{\{usr\}, \{Records\}, \{opr\}\}, \{permit\}, \{CombiningAlg\})$.*

Where "*isPrimaryDoctor*" is the rule conditions. In this case, only healthcare provider who hold a role as "*isPrimaryDoctor*" are permit to access the object *Records*.

Definition 4. *A policy set (PS) is a container of one or more policies. PS may contain other policy sets, policies or both.*

An example of defined access control policies as following:

Policy 1 (Role separation of duty (SoD) constraints)**.** User to role assignments should respect SoD where number of roles from SoD assigned to the same user can not exceed N. $RSoD \subseteq (2^R \times N)$ is collection of pairs (rs,n) in RSoD, where, rs is a role set, $m \subseteq rs$ and n is a integer number ≥ 2, with the property that no user is assigned to n or more roles from the set rs in each $(rs,n) \in RSoD$. Formally: $\forall(rs,n) \in RSoD, \forall m \subseteq rs : |m| \geq n \Rightarrow \bigcap_{r \in m} authorized\text{-}usr_{role}(r) = \emptyset$.

Policy 2 (Team role separation of duty (SoD) constraints)**.** A user in one team must be assigned to exactly one team role. Formally: $\forall t \in T, \forall usr \in USR : usr \in team_{member}(t) \Rightarrow |(authorized\text{-}usr_{tr}(usr,t))| = 1$.

Policy 3 (Role cardinality constraints)**.** User to role assignments must respect the cardinality constraints where the number of authorized users for any role does not exceed the authorization cardinality ($card_{role}(r)$) of that role. Formally: $\forall r \in R : |authorized\text{-}usr_{role}(r)| \leq card_{role}(r)$.

Policy 4 (Team cardinality constraints)**.** The number of authorized users for any team (t) does not exceed the cardinality ($card_{team}(t)$) of that team. Formally: $\forall t \in T : |team_{member}(t)| \leq card_{team}(t)$.

Policy 5 (Team role cardinality constraints)**.** The number of authorized users for any team role does not exceed the cardinality ($card_{team\text{-}role}(tr)$) of that role. Formally: $\forall tr \in TR : |team\text{-}role_{member}(tr)| \leq card_{team\text{-}role}(tr)$.

Policy 6 (Access)**.** Object "Records" can be accessed only by users with associated roles. Formally: $\forall usr \in USR, \forall r \in R, \forall per \in PER : usr \in authorized\text{-}usr_{role}(r) \wedge per \in assigned\text{-}role_{per}(r) \Rightarrow can\text{-}access(usr, Records)$, where $per = (Records, opr)$.

Definition 5. *An access query (Q) is a request by a user to perform an operation on an object. Formally:* $Q = (usr_{rq}, obj_{rq}, opr_{rq})$ *where,* $usr_{rq} \in USR$, $obj_{rq} \in OBJ$ *and* $opr_{rq} \in OPR$.

A response to access query Q is an access response $RS = \{D\}$ which is a decision D to the access query against the defined policy and rules (definitions 2 and 3) in policy set PS.

Definition 6. *Access control decision function (df) is defined as if condition defined by rules in access policy is evaluated as true or false (Algorithm 1), then there is decision* $D = \{permit, deny, indeterminate\}$*, according to the rules in the policy. Formally:* $df : Q \times PS \rightarrow D$.

Algorithm 1 evaluate the access request against the rules defined in policy. It takes *rule* and Q as an input and the result of evaluation would be the rule decision. The algorithm begins by checking the rule applicability against the request (Line 4). This is done by comparing the request Q against the rule target. The target contains a set of attributes and their values for matching the usr, object and operation, to check if the given rule is applicable to the request. If the target match, the rule condition is evaluated (line 6), otherwise, a discovery that the rule does not apply to the request "Indeterminate" (line 16) is returned to the policy df. Incorrect evaluation results in an error "Indeterminate" (Line 7), while correct evaluation of a condition returns the effect of the rule (line 9-13) which can be can be either a permission or denial of request Q. In the case there exists a *ruleEff=permit* and a *ruleEff=deny*, the decision will be taken according to the policy combining algorithm (*CombiningAlg*). For example, if $CombiningAlg = Deny\text{-}Overrides$, decisions will be combines in such a way that if any *ruleEff* is a deny, then that decision is deny.

Algorithm 1: Rule evaluation algorithm ($RuleEva(rule, Q)$).

Input *rule*, *target* and Q
Output Rule decision

1: $rule = \{\{ruleId\}, \{ruleCon\}, \{target\}, \{ruleEff\}, \{CombiningAlg\}\}$
2: $target = \{\{usr\}, \{obj\}, \{opr\}\}$
3: $Q = \{usr_{rq}, obj_{rq}, opr_{rq}\}$
4: **if** $((usr_{rq} \cap usr) \neq \emptyset) \wedge (obj_{rq} \cap obj) \neq \emptyset) \wedge (opr_{rq} \cap opr) \neq \emptyset)$ **then** ▷ Target is applicable
5: **for** $(i \in rule)$ **do**
6: **if** (i.ruleEff= false) **then**
7: $RuleDecision = Indeterminate$
8: **else**
9: **if** (i.ruleEff= Deny) **then**
10: $RuleDecision = Deny$
11: **end if**
12: $RuleDecision = permit$
13: **end if**
14: **end for**
15: **else**
16: Indeterminate ▷ Target is indeterminate
17: **end if**

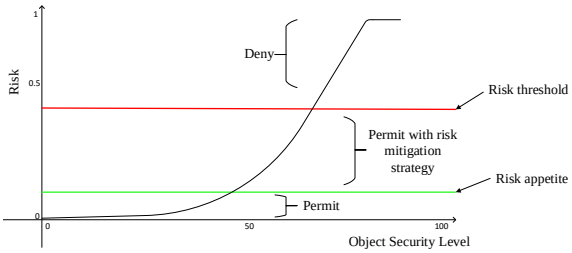


Figure 1: WBAC risk scale.

4 RISK AWARE FRAMEWORK

In this section, we present our risk framework. First, we present the risk appetite and risk threshold and then object security labels and user trust level. Followed by, user trust calculation and risk value calculation.

4.1 Risk Appetite and Thresholds

In the decision function (definition 6), the static decision (permit, deny, indeterminate) would be replaced by a dynamic access decision based on the risk value. Figure 1 presents the risk scale of our model, where the risk curve is divided into three bands.

The first band (risk appetite boundary) is associated with a decision permission because there is no risk or the risk is very low and an entity (e.g., organization) is willing to take the risk in anticipation of a reward.

The second band (risk threshold boundary) is assigned a decision permission with risk mitigation plans (e.g., risk acceptance (Rittenberg and Martens, 2012), risk transfer and risk avoidance (Stoneburner et al., 2002)). Below that risk threshold (between the risk appetite boundary and risk threshold boundary) is the risk tolerance, which is the amount of risk that an entity will accept or is willing to withstand. Risk threshold is measured along the level of uncertainty or level of impact at which an entity may have a specific interest. Below that risk threshold, the entity will accept or withstand the risk. Above that risk threshold, the entity will not tolerate the risk. Risk threshold varies from one entity to another depending on the risk level an entity is willing to take.

The third band is associated with the deny decision because the risk is too high. In this case, it is not desirable to prevent a healthcare provider from accessing an object as this could cause greater damage to the patient than the risk of accessing the patient's records. This is due to requirement of "nothing must interfere with the delivery of care" (Salim et al., 2011; Rostad et al., 2007). But the object owner or system administrator (e.g. security administrator) must be notified of the risk and the access request will require evaluating the patient's (object owner) consent. Therefore, the problem of low risk detectability can be examined and the object owner or system administrator may carry out an investigation and discover the facts of access.

In a case of the risk value is greater than the risk appetite and risk threshold, the healthcare provider's access request to the object will not be denied completely, but mitigation plans can be put in place to reduce the risk. An example of a mitigation plan is an additional evaluation layer such as a purpose-based access control policy to solve conflicting analysis (Wang et al., 2010). The purpose of information access could be associated with the access request to specify the intention of the access request.

4.2 Object Security Labels

Every object in our model is associated with a security label which represents the level of object protection. Based on the level of sensitivity, the object can be classified in numerous ways. For example, one of the following labels is assigned to the object are top secret, secret, confidential and unclassified (Stewart et al., 2015).

Definition 7. *The object security labels denoted by security-level*(obj) *and is defined in the interval* $[0,1]$*, where 0 means the required level of protection is very low and 1 means the level of protection required is very high.*

Object security level is assigned to an object by the object owner. It is understood that healthcare providers need varying degrees of patient information to perform their job duties (Salim et al., 2011). Thus, object classification require a great deal to set a security levels to patient data as well as define what access privileges could lead to potential insider threats and compromise patient data. However, we believe that assigning a security level to a medical record based on the sensitivity of information is doable. Here, we assume that the object owner decides on the object's security level based on its sensitivity. For example, psychotherapy notes (i.e., notes recorded by a health care provider who is a mental health professional (US Department of Health and Human Services et al., 2014)) could have a higher security level comparing to patient personal information such as phone number and address. In our access control model (section 3), and according to our object classification, such an object (psychotherapy notes) will be accessible to treatment

team only if treatment team decide such information is necessary for patient treatment.

4.3 User Trust Level

User trust level has been defined by Mayer et al., 1995, as a function of trustee's behavior that includes its ability, benevolence and integrity of the trustor's propensity to trust. In this study, we assume that, each user (healthcare provider) is associated with a trust level that represents the level of user clearance by the organization (e.g., hospital and clinic) that owns the data.

Definition 8. *The trust for a user (usr) towards an object (obj) is denoted by trust-level(usr) and is defined in the interval* $[0,1]$*, where 1 means the user is fully trusted and 0 means the user is totally untrusted.*

User trust level may be computed either statically or dynamically. Static computation refers to user attributes. For example, the trustworthiness level of cardiologists in the cardiology department may be higher than that of the nurses who work in the same department. Dynamic computation refers to users' access history (Cheng et al., 2007), behavior (Shaikh et al., 2012; Shaikh et al., 2011; Baracaldo and Joshi, 2013) and value of user reputation (Bijon et al., 2013).

As we mentioned early (section 2), Shaikh and others in (Shaikh et al., 2011; Shaikh et al., 2012) proposed a dynamically user trust calculation model based on rewards and penalty points. The authors assumed that, on the one hand, if the access request to an object is successfully redeemed, an obligation server will assign rewards points to the user with a respect to the object. On the other hand, if the access request is not successfully redeemed due to any problem, the obligation server will assign penalty points.

In this study, we modify the proposed method of calculating rewards and penalty histories by assigning different weight values based on the oldness of their occurrence. That is, the older the action that lead to a specific reward/penalty, the smaller the weight given to that reward/penalty. This is done by incorporating a forgetting factor (Hayes, 2009) which decays exponentially with the time lag between the instance of calculating the trust level and the reward/penalty action occurrence.

4.4 User Trust Calculation

For each user, we calculate a trust level value $trust\text{-}level(usr)$ with a respect to an object. The calculation is done based on user previous trust level value TL_{value}, rewards RE and penalty PE points with a regards to the time τ as in equation 1. In case of new user (user has no TL_{value}) or if neither penalties nor rewards are available, then the user trust will be assigned to the lower trust level TL_{low}.

$$trust\text{-}level(usr) = TL_{value} + \left(\frac{\sum_{i=1}^{n} RE(i) \cdot \lambda_{RE}^{\tau_i - \tau_0} - \sum_{i=1}^{n} PE(i).\lambda_{PE}^{\tau_i - \tau_0}}{n} \right) \quad (1)$$

Where, i is an event index, $\lambda_{RE} \leq 1$ is rewards forgetting factor, $\lambda_{PE} \leq 1$ is penalty forgetting factor, τ_i is a time of the event occurrence i, τ_0 is the current time of calculating the user trust level, and n is the number of events.

4.5 Risk Value Calculation

Risk is defined as the probability that a hazardous situation (threat) will occur and the impact of successful violation. When designing a system, it is necessary to understand the risk exposure level and ensure it is within risk appetite. The National Institute of Standards and Technology (NIST) (Stoneburner et al., 2002) calculate the risk based on threat probability and the impact. In this paper, we determine the threat probability based on the number of rewards and penalty points and the impact is defined based on the object security label (section 4.2). For example, if the object security label is high (e.g., secret) than the risk threshold is high and the impact will be high (this depends on the user trust level).

Before we start explain how we calculate the risk value, we first, give an auxiliary predicates we need for our risk assessment. The following functions are defined:

- $risk\text{-}appetite(opr,obj)$: The risk appetite value used to determine the amount or volume of risk that an organization or individual will accept and is defined in the interval $[0,1]$.
- $risk\text{-}threshold(opr,obj)$: The risk threshold value is used to determine whether the risk is acceptable to an organization and is defined in the interval $[0,1]$ which

Given rewards and penalty points, we calculate the risk associated with an access request Q based on the user trust level ($trust\text{-}level(usr)$) and the object security level ($security\text{-}level(obj)$) as shown in equation 2.

$$rv(usr,obj) = (sigm(security\text{-}level(obj) - trust\text{-}level(usr)). \quad (2)$$

Where, $sigm(x) = \frac{1}{1+e^{-x}}$.

We believe there are many ways to calculate the risk value. Here, we chose the sigmoid function

(equation 2) as an activation function to weigh the risk sum of the object security level and user trust level and present the output as a sigmoid curve. The sigmoid function is real-valued and differentiable maps binary events to real-valued (probabilistic) curves. It has a non-negative or non-positive first derivative, which makes calculating and learning the weights of a risk easier. It is also a very popular function used in machine learning, artificial neural networks (Basheer and Hajmeer, 2000) and modeling risk management (Cheng et al., 2007), because it satisfies a property between the derivative and itself, such that it is computationally easy to perform.

4.6 Risk-aware Access Decision Mechanism

After calculating the trust level of user and the risk value, The risk parameter is added to the decision function (definition 6). It is said a threat exists if a $usr \in USR$ can access an $obj \in OBJ$ such that $rv(usr,obj) \geq risk\text{-}appetite(opr,obj)$.

Definition 9. *Risk-Aware Access Decision Mechanism is defined as if condition defined in rules is evaluated as true and the risk value $rv(usr,obj)$ is less than $risk\text{-}appetite(opr,obj)$, the usr is permitted, otherwise, denied. Formally:*

$$rdf : Q \times PS \times rv(usr,obj) \rightarrow D$$

$$rdf = \begin{cases} \text{Permit} & \text{if } Case1. \\ \text{Mitigated} & \text{if } Case2. \\ \text{Deny} & \text{if } Case3. \\ \text{Indeterminate} & Otherwise. \end{cases}$$

Where, PS and $rv(usr,obj)$ are the policy set (definition 2) and risk value date sets, respectively, related to object obj and

- Case1= $\exists$ $rule$ $[rule.ruleEff == permit] \wedge rv(usr,obj) \leq risk\text{-}appetite(opr,obj)$.
- Case2= $\exists$ $rule$ $[rule.ruleEff == permit] \wedge risk\text{-}threshold(opr,obj) \leq rv(usr,obj) \geq risk\text{-}appetite(opr,obj)$.
- Case3= $\exists$ $rule$ $[rule.ruleEff == deny] \vee rv(usr,obj) \geq risk\text{-}threshold(opr,obj)$.

We say, if the risk value given by $rv(usr,obj)$ fails between the first and second bands (Figure 1) and there exists rules (Algorithm 1) permit the request, then user (usr) is permitted to perform operation (opr) on object (obj) with risk value ($rv(usr,obj)$) and risk mitigation plans, otherwise, the access request is denied. Unless the request is approved by the object owner or system administrator. In deny case, we also assume it might be rules in access policy deny the access. Indeterminate means there is no rules applicable to the request.

5 RISK ASSESSMENT IN ACCESS DECISIONS

In this section, first, we consider a usage scenario and then we give examples of how the access decision is made according to user trust level and risk value.

5.1 Usage Scenario

Considering the following case study (adapted from (Zhang and Liu, 2010)). Four healthcare providers (*Dean*, *Bob*, *Cara* and *Alex*) are working on a case (treatment of a patient *Alice*). In our model, we assume *Dean* is assigned the primary doctor role and he serves as the group manager. He is responsible for initiating the work (*Alice*'s treatment case) and choosing the practitioners (team of doctors) who may be required to attend *Alice*'s consultation and treatment. *Dean* assigns users (*Bob*, *Cara* and *Alex*) to the team. The members join the team and are assigned team roles based on the required job function.

We assume that *Dean* decides who should access what based on the required job. Table 1 presents the policy data used as input for our proof of concept.

As shown in Table 1, we assume that objects (*Alice*'s health recors) are classified to $obj_{a(Alice)} = \{File_1, ..., File_n\}$, where $file_i$ is private information such as *Alice*'s personal data and other information not related to the current case (e.g sexually transmitted diseases (STD)), and $obj_{b(Alice)} = \{File_1, ..., File_n\}$, where $file_i$ is information related to *Alice*'s current case, such as her old medical records. The security level of each object are assigned as shown in Table 2.

Bob, *Cara* and *Alex* are healthcare professionals who join Alice's treatment team and are assigned to team role. We assume, if all conditions (e.g. separation of duty (SOD) are true, then role R and TR are assigned to users and the access state Γ (definition 1) will be updated as shown in example 2.

Example 2. ***Access State*** *(γ_1): Consider Alice's case, the initial state denoted by γ_1 is the formal model assignment presented in Table 1 as follows:*

$USR = \{Dean, Bob, Cara, Alex\}$.
$R = \{Primary\text{-}doctor, General\text{-}practitioner, Gastroenterologist, Medical\text{-}coordinator\}$.
$TR = \{tr_a, tr_t, tr_m\}$.
$T = \{t_1\}$.

Table 1: Tabular structure of policy data.

Subject	Job Function	Team Role	Object Type	Action	Permission
Dean	Primary Doctor	Role	$OBJ_{A(Alice)}$ and $OBJ_{B(Alice)}$	Read/write	Permit
Bob	General practitioner	Action	$OBJ_{A(Alice)}$ and $OBJ_{B(Alice)}$	Read	Permit
Cara	Gastroenterologist	Thought	$OBJ_{B(Alice)}$	Read	Permit
Alex	Medical coordinator	Management	$OBJ_{B(Alice)}$	Read	Permit

Table 2: Assumptions of object security levels.

Object	SL	Description
obj_A	0.8	*Alice*'s personal information (e.g. name, phone number, STD) and other EHRs that are not related to Alice's current case
obj_b	0.5	Alice's medical records that are related to *Alice*'s current case

$W = \{w_1\}$.
$OBJ = \{obj_a, obj_b\}$.
$OPR = \{Read, Write\}$.
$PER = \{(read, obj_a), (write, obj_a), (read, obj_b), (write, obj_b)\}$.
where,
$USR\text{-}R\text{-}A = \{(Dean, Primary\text{-}doctor)\}$.
$PER\text{-}R\text{-}A = \{(Primary\text{-}doctor, (read, obj_a)), (Primary\text{-}doctor, (write, obj_a)), (Primary\text{-}doctor, (read, obj_b)), (Primary\text{-}doctor, (write, obj_b))\}$.
$USR\text{-}T\text{-}A = \{(Alex, t_1), (Cara, t_1), (Bob, t_1)\}$.
$TM\text{-}TR\text{-}A = \{(Bob, t_1, tr_a), (Cara, t_1, tr_t), (Alex, t_1, tr_m)\}$.
$PER\text{-}TR\text{-}A = \{(tr_a, (read, obj_a)), (tr_a, (write, obj_a)), (tr_a, (read, obj_b)), (tr_t, (read, obj_b)), (tr_m, (read, obj_b))\}$.
$T\text{-}W\text{-}A(t_1, w_1) = \{(w_1, t_1)\}$.
$risk\text{-}appetite(usr, obj) = 0.18$ (assumption).
$risk\text{-}threshold(usr, obj) = 0.45$ (assumption).

5.2 Proof of Concept

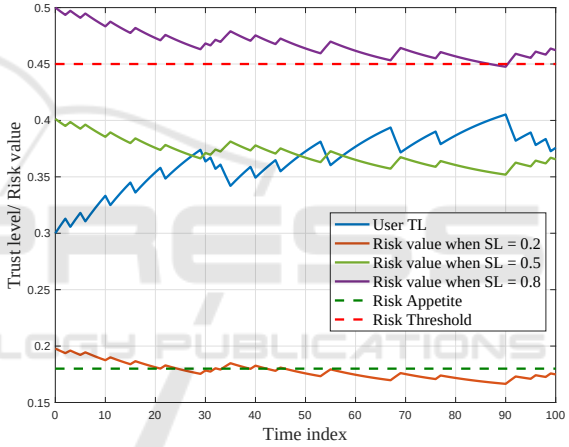
In this proof of concept, we consider the access state γ_1 which is a particular model assignment to a given WBAC system. An access state $\gamma \in \Gamma$contains all the information necessary to make access control decisions for a given time.

Example 3 (User permission with risk assessment:). *Let assume an example where Cara requests read access to file in $obj_{b(Alice)}$ and let $q = (Cara, obj_b, read)$ where q is the access request (definition 5).*

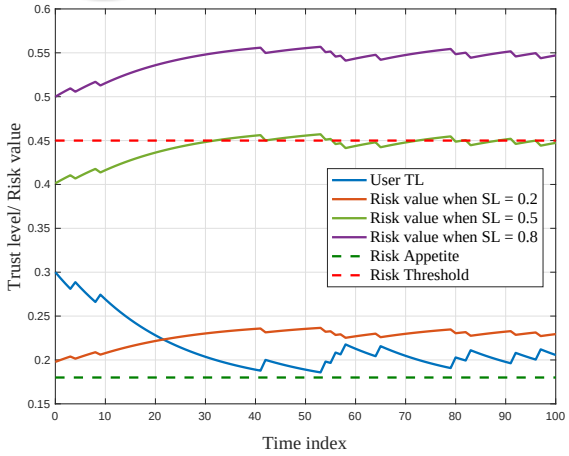
Table 2 shows that $security\text{-}level(obj_b) = 0.5$ and access state γ_1 (Example 2) shows that *Cara* is a member of team t_1 and she is assigned to team role tr_t. According to γ_1, we could say that the system dose not violate the any constraints (e.g., SoD and cardinality constraints) and based on the access decision function df (Definition 6), *Cara* could access and preform read operation on a file in $obj_{b(Alice)}$ as she is a member of a team and she hold a team role (policy 6).

Considering the user trust level (Figures 2) as the basic criterion for conducting risk assessment, we could see how trust level of user and risk values increase and/or decreases with the change in user behavior. For *Cara*' request, it can be concluded that permitting *Cara* to read obj_b has low risk (Figure 2(a)) comparing with her trust level which was calculated according to her history of rewards and penalty points.

On the one hand, in figure 2(a), we assumed



(a) Trust level and risk value in case of 20% misbehaving user



(b) Trust level and risk value in case of 90% misbehaving user

Figure 2: Risk exposure using our model.

that *Cara* is 80% behaving according to rules and she only violates the system rules and policy about 20% (i.e., 20% misbehaving user). Figure 2(a) indicates that the risk value $rv(Cara, obj_{b(Alice)})$ is greater than $risk\text{-}appetite(usr, obj)$ and less than $risk\text{-}threshold(usr, obj)$. Therefore, user *Cara* is permitted to perform operation *read* on object $obj_{b(Alice)}$ with a risk value of $\simeq 0.4$ (function 2). This happens when $trust\text{-}level(Cara) \approx 0.3$.

On the other hand, figure 2(b) shows the case when *Cara* is 90% misbehaving user. As shown in the Figure 2(b), *Cara* is posing a high risk towards $obj_{b(Alice)}$ since the $trust\text{-}level(Cara) \approx 0.2$ is less than $security\text{-}level(obj) = 0.5$ then the $rv(usr, obj) \simeq 0.45 \geq risk\text{-}threshold(usr, obj)$. According to the principle presented in (Cheng et al., 2007) and Figure 2(b), the threat always increases as the difference between the object security level and user trust level increases and and vice versa.

As we mentioned early (section 4.1) that we can not prevent a healthcare provider from accessing an object during patient treatment. However, in our access control model (section 3), the risk that *Cara* poses to the object has been mitigated. Upon conducting risk assessment of *Alice*'s object, it is noted that *Cara* poses a threat to this file. Therefore, the problem of low risk detectability is examined and *Alice* or the system administrator has discovered the fact of an access (*Alice*'s treatment).

6 DISCUSSION AND CONCLUSIONS

6.1 Discussion

In our access control model, a realistic way of handling collaboration risk is to minimize the discrepancy between the granted and required access. This is done by utilizing resource classification in organizing shared resources and team roles. *Dean* could retrieve all resources that he thinks team members does not need to for *Alice*'s treatment. *Dean* must also receive *Alice*'s consent (explicitly or implicitly) regarding treatment team formation and which objects according to "need-to-know" principle (Ferraiolo et al., 2001) and "minimum necessary" standard (Agris, 2014) can be shared with the team. This would be done to satisfy the requirement of "selective relevancy" where *Alice* would be able to withhold information that remains confidential.

In reality, patients cannot decide what information is needed to complete their treatment, therefore doctors must choose the required information themselves and the patient has to agree (Salim et al., 2011; Rostad et al., 2007). In any case, the patient is aware of the threat posed by the healthcare provider to the objects and the patient can refuse to share objects with the healthcare provider. Although the patient may refuse to share, the risk of refusing to share objects with healthcare providers poses a greater risk than the risk of sharing the objects. Thus, in our access control model, the extent of collaborative access has been elevated and then audited the increased responsibilities of an insider pre-justifying the extent request, and an authorizing entity (patient or primary doctor) granting the access possibly for a specified ahead access time which would be done by the work component. This would make an insider threat more detectable. In addition, the activity of a healthcare provider who joined the team and was granted such access extent can be logged and audited for intrusion detection if there is contradiction of the pre-justified request for the extent. In a case where the risk is substantial and non-acceptable ($rv(Cara, obj_b) \geq risk\text{-}threshold(usr, obj_b)$), the policy must be amended to allow such additional constraints.

6.2 Conclusions

The motivation behind creating a risk assessment framework for our enhances access control model is to help improve system security in terms of protecting healthcare information from insider threats, such as patient data disclosure and/or unauthorized access or modification by insiders. We, first, briefly talked about our access control model (WBAC). Second, we presented the concept of user trust level and object security level. Also, we introduced the idea of assigning rewards and penalty points for a users and how this rewards and penalty points are used to calculate the user trust level. The main goal of our risk assessment framework is to evaluate the risks associated with access requests and based on the risk value, an access decision is made.

In the future, we plan to extent the model by considering other factors such as cost and impact of permission, risk associated with role/ team role assignments as well as directly compared our model with exciting models. Also, we plan to evaluate the validity of the scheme to provide solutions to improve healthcare quality, provide access to a high-quality healthcare system and support close cooperation between healthcare professionals and care providers from different organization.

REFERENCES

Abomhara, M. and Køien, G. M. (2016). Towards an access control model for collaborative healthcare systems. In *HEALTHINF'16, 9th International Conference on Health Informatics*, volume 5, pages 213–222.

Abomhara, M., Yang, H., Køien, G. M., and Lazreg, M. B. (2017). Work-based access control model for cooperative healthcare environments: Formal specification and verification. *Journal of Healthcare Informatics Research*, pages 1–33.

Agris, J. L. (2014). Extending the minimum necessary standard to uses and disclosures for treatment: Currents in contemporary bioethics. *The Journal of Law, Medicine & Ethics*, 42(2):263–267.

Baracaldo, N. and Joshi, J. (2013). An adaptive risk management and access control framework to mitigate insider threats. *Computers & Security*, 39:237–254.

Basheer, I. and Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1):3–31.

Bell, D. E. and LaPadula, L. J. (1975). Computer security model: Unified exposition and multics interpretation. *MITRE Corp., Bedford, MA, Tech. Rep. ESD-TR-75-306, June*.

Bijon, K. Z., Krishnan, R., and Sandhu, R. (2013). A framework for risk-aware role based access control. In *Communications and Network Security (CNS), 2013 IEEE Conference on*, pages 462–469. IEEE.

Chao, C.-A. (2016). The impact of electronic health records on collaborative work routines: A narrative network analysis. *International journal of medical informatics*, 94:100–111.

Cheng, P.-C., Rohatgi, P., Keser, C., Karger, P. A., Wagner, G. M., and Reninger, A. S. (2007). Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In *2007 IEEE Symposium on Security and Privacy (SP'07)*, pages 222–230. IEEE.

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., and Chandramouli, R. (2001). Proposed nist standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 4(3):224–274.

Hayes, M. H. (2009). *Statistical digital signal processing and modeling*. John Wiley & Sons.

Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., and Scarfone, K. (2014). Guide to attribute based access control (abac) definition and considerations. *NIST Special Publication*, 800:162.

Li, N., Wang, Q., Qardaji, W., Bertino, E., Rao, P., Lobo, J., and Lin, D. (2009). Access control policy combining: theory meets practice. In *Proceedings of the 14th ACM symposium on Access control models and technologies*, pages 135–144. ACM.

Probst, C. W., Hunker, J., Gollmann, D., and Bishop, M. (2010). *Insider Threats in Cyber Security*, volume 49. Springer Science & Business Media.

Reitz, R., Common, K., Fifield, P., and Stiasny, E. (2012). Collaboration in the presence of an electronic health record. *Families, Systems, & Health*, 30(1):72.

Rittenberg, L. and Martens, F. (2012). Enterprise risk management: understanding and communicating risk appetite. *COSO, January*.

Rostad, L., Nytro, O., Tondel, I., and Meland, P. H. (2007). Access control and integration of health care systems: An experience report and future challenges. In *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, pages 871–878. IEEE.

Salim, F., Reid, J., and Dawson, E. (2010). Authorization models for secure information sharing: A survey and research agenda. *The ISC International Journal of Information Security*, 2(2):69–87.

Salim, F., Reid, J., Dawson, E., and Dulleck, U. (2011). An approach to access control under uncertainty. In *Availability, reliability and security (ARES), 2011 Sixth International conference on*, pages 1–8. IEEE.

Shaikh, R. A., Adi, K., and Logrippo, L. (2012). Dynamic risk-based decision methods for access control systems. *computers & security*, 31(4):447–464.

Shaikh, R. A., Adi, K., Logrippo, L., and Mankovski, S. (2011). Risk-based decision method for access control systems. In *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on*, pages 189–192. IEEE.

Shoniregun, C. A., Dube, K., and Mtenzi, F. (2010). *Electronic healthcare information security*, volume 53. Springer Science & Business Media.

Stewart, J. M., Chapple, M., and Gibson, D. (2015). *CISSP (ISC)2: Certified Information Systems Security Professional Official Study Guide*. John Wiley & Sons, Seventh Edition edition.

Stoneburner, G., Goguen, A. Y., and Feringa, A. (2002). Special publication 800-30: risk management guide for information technology systems. *National Institute of Standards & Technology*.

US Department of Health and Human Services et al. (2014). Hipaa privacy rule and sharing information related to mental health.

Wang, H., Sun, L., and Varadharajan, V. (2010). Purpose-based access control policies and conflicting analysis. In *Security and Privacy–Silver Linings in the Cloud*, pages 217–228. Springer.

Zhang, R. and Liu, L. (2010). Security models and requirements for healthcare application clouds. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 268–275. IEEE.