UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

ANALYSE DE RISQUE DANS LES SYSTEMES DE CONTRÔLE D'ACCES

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE MAITRISE EN INFORMATIQUE

PAR

Sourour JEMILI

Préparé au sein du

Laboratoire de Recherche en Sécurité Informatique

Soutenue en 2013 à l'UQO devant le jury composé de :

Président : ..... Michal Iglewski, Professeur à l'UQO

Membre : ..... Mohamed Mejri, Professeur à ULaval

Directeur de recherche : ..... Kamel Adi, Professeur à l'UQO

Co-directeur de recherche :..... Luigi Logrippo, Professeur à l'UQO

A mon père et ma mère,

A mon frère et mes sœurs,

A ma famille et tous mes amis,

en témoignage de mon grand amour…

« Patience ! Avec le temps, l'herbe devient du lait »

Proverbe chinois

« N'allez pas là où le chemin peut mener. Allez là où il n'y a pas de chemin et laissez une trace »

Ralph Waldo EMERSON

Remerciements

De nombreuses personnes m’ont aidé à la réalisation de ce travail. Je tiens à les remercier vivement.

J'adresse tout d'abord mes remerciements au Professeur Kamel ADI, pour m'avoir accueillie dans le laboratoire LRSI et pour ses précieuses directives et son soutien perpétuel. Qu'il trouve ici le témoignage de ma gratitude et de mon grand respect.

Un grand merci aussi au Professeur Luigi LOGRIPPO, qui a également dirigé mes travaux de recherche. Ses remarques constructives m'ont conduite à aborder le sujet de ce mémoire avec une plus grande rigueur.

Monsieur le Professeur Michal IGLEWSKI me fait un grand honneur en acceptant de présider le jury de soutenance de cette maitrise. Qu’il en soit grandement remercié.

Je souhaite exprimer tous mes remerciements au Professeur Mohamed MEJRI pour l’honneur qu’il me fait en acceptant d'être membre externe dans ce jury.

Je remercie aussi tous les membres du laboratoire LRSI, qui m'ont donné l'occasion de présenter et discuter les différents rapports d'avancement de ce travail de recherche, qu'ils trouvent ici ma reconnaissance pour leur disponibilité et pour leur soutien continu.

Je tiens également à témoigner ma reconnaissance à tous mes enseignants, en particulier, ceux de l’Université du Québec en Outaouais pour l’effort qu’ils ont fait tout au long de mon cursus universitaire.

Je voudrais aussi remercier ma famille, en particulier ma mère, mon père, mon frère et mes sœurs, pour leur soutien incessant qui m'a été plus que précieux tout au long de mes études.

Oublier de remercier les chercheurs qui m'ont inspiré ou ceux qui m'ont servi de modèles serait inqualifiable, je tiens à accorder une pensée à tous ceux et celles, connus ou reconnus, inlassablement, qui apportent des idées nouvelles ou reformulent avec une pertinence différente et novatrice des concepts déjà populaires. Entre les lignes de leurs écrits, j'admire leur enthousiasme jamais défaillant, et envie leur générosité.

1 Sommaire

Chapitre 1: Introduction

> *Le présent chapitre est une introduction au projet de recherche. Nous y présentons des notions de sécurité informatique et du contrôle d'accès dans les systèmes d'information et précisons nos objectifs de recherche.*

La sécurité d'un système informatique a pour but la protection des ressources informatiques (incluant les données et les programmes) contre la divulgation, la modification ou l'utilisation non-autorisée, tout en garantissant l'accès pour les utilisateurs légitimes.

Généralement, la sécurité informatique vise trois principaux objectifs : l'intégrité, la confidentialité et la disponibilité des données. L'intégrité permet de déterminer si les données n'ont pas été altérées que ce soit de manière accidentelle ou intentionnelle. La confidentialité consiste à assurer que seules les personnes autorisées aient accès aux ressources protégés. La disponibilité permet de garantir l'accès aux informations et services.

Pour assurer la sécurité informatique, plusieurs techniques sont employées dont le contrôle d'accès, le chiffrement de l'information (cryptographie), la protection contre les signaux parasites compromettants (sécurité électronique), la protection contre les intrusions dans les logiciels, mémoires ou banques de données (sécurité logique) et la protection contre les accidents naturels et les actes malveillants (sécurité physique). Dans le cadre de notre travail, nous nous intéressons au contrôle d'accès.

Le contrôle d'accès [1] est un mécanisme grâce auquel un système de contrôle autorise ou interdit les actions demandées par des sujets (entités actives) sur des objets (entités passives).

Le contrôle d'accès renforce les bonnes propriétés de disponibilité, de confidentialité et d'intégrité de l'information. Il est généralement mis en œuvre par un moniteur qui contrôle le flux de transactions entre les utilisateurs et les ressources protégés auxquelles ces derniers essaient d'accéder.

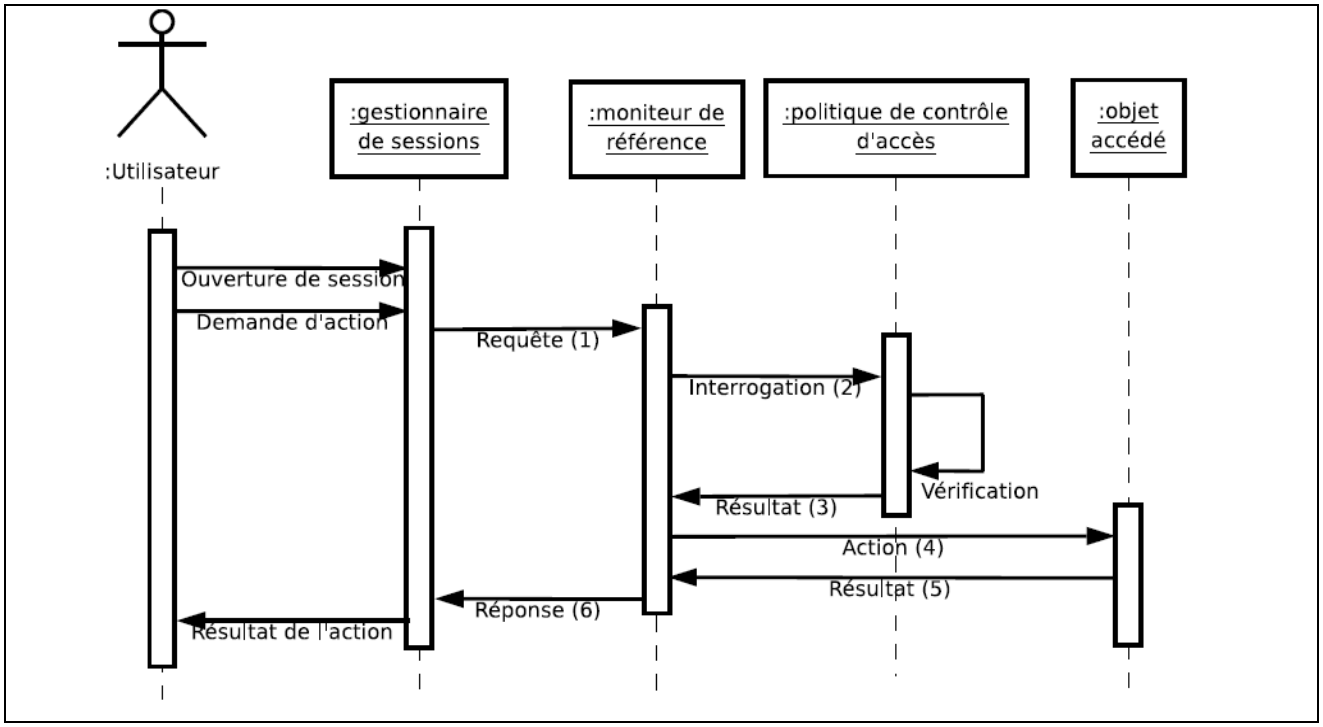


Figure 1 : Mécanisme de moniteur mis en œuvre pour réaliser le contrôle d'accès [2]

Lorsqu'un utilisateur demande un accès, le moniteur décide alors si cet accès est autorisé ou non en conformité avec la politique de contrôle d'accès en vigueur. Le principe de fonctionnement d'un moniteur est décomposable en six étapes (voir Figure 1) :

1. Envoi de la requête de l'utilisateur au moniteur.
2. Interrogation de la politique de contrôle d'accès.
3. Réponse du gestionnaire de la politique de contrôle d'accès.
4. Accès du moniteur à la ressource (exécution de la requête) si le gestionnaire de la politique de contrôle d'accès l'y autorise.
5. Retour du résultat de l'exécution de la requête vers le moniteur de référence.
6. Réponse à la requête de l'utilisateur.

Par ailleurs, tout système de contrôle d'accès doit être conforme à une politique de sécurité qui regroupe l'ensemble des lois, règles et pratiques qui gèrent l'accès, la distribution et la protection des ressources à l'intérieur du système. De plus, il est important qu'un système de contrôle d'accès soit conforme à un modèle de sécurité bien défini afin de faciliter les preuves sur la sécurité du système informatique. Le modèle étant une représentation formelle des politiques de sécurité et de leurs fonctionnements. Finalement, un système de contrôle d'accès doit toujours être

mis en œuvre par des mécanismes de sécurité qui définissent les fonctions de bas niveau (logiciels et matériels) permettant d'implémenter les contrôles imposés par la politique de sécurité.

Plusieurs modèles de contrôle d'accès sont décrits dans la littérature, parmi lesquels, nous pouvons citer :

- Le modèle de contrôle d'accès discrétionnaire ou DAC (Discretionary Access Control) où la politique d'accès est laissée à la discrétion des utilisateurs du système, i.e. : chaque utilisateur peut octroyer des privilèges sur les ressources qui sont sous sa juridiction.
- Le modèle de contrôle d'accès obligatoire ou MAC (Mandatory Access Control) où les politiques de sécurité imposent que les décisions de protection ne doivent pas être prises par le propriétaire des objets concernés, et doivent lui être imposées par le système.
- Le modèle de contrôle d'accès basé sur les rôles ou RBAC (Role-Based Access Control), où les privilèges sont octroyés à des rôles d'une organisation et les utilisateurs sont eux-mêmes assignés aux rôles.

Il est utile de signaler que de nos jours, le modèle le plus couramment utilisé est le modèle RBAC car il est mieux adapté aux modèles organisationnels contemporains. Malheureusement, avec l'émergence de l'informatique ubiquitaire et ses environnements hautement dynamiques, ce modèle a atteint ses limites et il est désormais, de plus en plus, sujet à des attaques de sécurité comme, par exemple, les attaques par violation d'identité permettant à un utilisateur d'user incorrectement de privilèges accordées à un rôle. Dans ce contexte, l'enrichissement du modèle par un calcul judicieux des risques d'utilisations incorrectes des ressources à protéger permettra certainement de renforcer la sécurité du modèle.

Dans ce mémoire, nous proposons un nouveau modèle de contrôle d'accès basé sur les rôles avec gestion du risque. Dans notre modèle, l'évaluation de risque devient un ingrédient majeur dans le processus de contrôle d'accès et le contrôle d'accès est en permanence ajusté par rapport à ces valeurs de risques qui sont calculées dynamiquement et en temps réel et en prenant en compte les informations contextuelles.

Chapitre 2 : État de l'art

Après avoir précisé la place du contrôle d'accès au sein de la problématique générale de la sécurité des systèmes d'information, nous présentons, dans ce chapitre, l'état de l'art relatif à notre sujet. Ainsi, nous détaillerons un échantillon représentatif des recherches sur les modèles de contrôle d'accès en donnant, à chaque fois, les motivations sous-jacentes à l'introduction de ces modèles.

2.1 Introduction

La mise en place d'une politique de contrôle d'accès dans un système est un chantier important pour une organisation. Une telle entreprise est à la croisée de l'informatique, de la gestion des ressources humaines et de la qualité de service.

De nos jours nous parlons, de plus en plus, de « l'informatique ubiquitaire », modèle introduit par Weiser en 1991 [3] et permettant de réaliser l'intégration transparente des environnements numériques au monde physique. Ce modèle permet à un utilisateur d'accéder à l'espace d'information de façon implicite comme résultat par ses actions dans l'environnement réel. Weiser considère que l'analyse des intentions des utilisateurs mobiles par la perception de leur environnement doit permettre au système mobile de suggérer des «réponses» adaptées. Comme l'utilisateur mobile est, par nature, soumis à des changements fréquents (changement de contexte d'utilisation, changement des dispositifs d'interaction, etc.), la définition et la modélisation d'un «contexte» spécifique est nécessaire afin de permettre de :

- percevoir la mobilité (le changement);
- s'adapter aux variations dans les conditions d'utilisation du système; et
- apporter des réponses (plus) appropriées

Par conséquent, les problèmes de sécurité dans l'informatique ubiquitaire deviennent beaucoup plus complexes par rapport aux environnements traditionnels, et les systèmes de contrôle d'accès mis en place doivent faire face à ces nouveaux défis.

2.2 Environnements d'informatique ubiquitaire et le contrôle d'accès

Il y a plusieurs problèmes de sécurité qui doivent être abordés dans les applications de l'informatique ubiquitaire [4]. En particulier, nous présentons, ci-après, les conditions essentielles à l'élaboration d'une architecture de contrôle d'accès sécurisée et flexible.

1. Accès basé sur les informations contextuelles.

Les modèles de contrôle d'accès existants sont majoritairement développées dans le contexte des systèmes traditionnels distribués et multiutilisateurs qui spécifient les accès autorisés pour les utilisateurs aux ressources du système et considèrent que les utilisateurs et les ressources sont statiques (ne changent pas dans le temps). Or, dans les environnements ubiquitaires les sujets et les ressources peuvent entrer et sortir, dynamiquement, de l'espace système rendant ainsi les modèles traditionnels inadéquats.

2. Accès pour différents utilisateurs et périphériques.

L'informatique ubiquitaire peut également être utilisée pour des applications non- techniques. Les utilisateurs dans cet environnement peuvent ne pas avoir une formation spéciale et peuvent préférer la facilité de l'utilisation des appareils.

3. Accès dans un environnement collaboratif.

L'informatique ubiquitaire est souvent utilisée pour des services de collaboration, où de nombreux utilisateurs travaillent ensemble pour accomplir une tâche. Un modèle de contrôle d'accès dans les environnements ubiquitaire doit fournir une structure de collaboration sécurisée pour les utilisateurs qui partagent des autorisations pour la tâche de collaboration.

4. Administration décentralisée.

Un système local dans l'informatique ubiquitaire n'est généralement qu'une partie d'un système plus vaste. Les politiques de contrôle d'accès doivent alors être organisées de manière décentralisée.

Les mécanismes traditionnels de contrôle d'accès sont insensibles au contexte. Ils exigent une infrastructure lourde et statique d'authentification. Il faut donc adapter ces systèmes aux environnements ubiquitaires. Les recherches récentes sur le contrôle d'accès sont principalement

orientées vers des systèmes basées sur le contexte et le rôle. Certaines recherches utilisent la confiance comme composante fondamentale pour le contrôle d'accès. D'autres, combinent la confiance avec le risque pour créer des services de sécurité renforcés mieux adaptés aux environnements ubiquitaires. Le risque étant souvent défini comme une combinaison de la probabilité et de la gravité d'un accident.

2.3 Techniques d'analyse de risque

L'évaluation des risques est un moyen permettant de fournir aux décideurs les informations nécessaires pour comprendre les facteurs qui peuvent influer négativement sur les conséquences et faire, ainsi, des jugements corrects pour réduire ces risques. C'est une étape importante dans le processus de gestion des risques, et aboutit à la détermination de la valeur quantitative ou qualitative des risques liés à des situations concrètes et des menaces reconnues. Le processus d'évaluation des risques comprend généralement :

1. Identification des menaces qui pourraient affecter les opérations critiques.
2. Estimation de la probabilité que ces menaces se matérialisent sur la base des informations historiques et des connaissances basées sur des jugements.
3. Identification et hiérarchisation de la criticité (degré de risque) des opérations de sorte que si une menace se matérialise, il est possible de déterminer quelles sont les opérations les plus importantes qui y ont contribuées.
4. Estimation des pertes ou des dommages au cas où une menace se matérialise, y compris les frais de recouvrement.
5. Définition de mesures rentables pour atténuer ou réduire les risques. Ces mesures comprennent la mise en œuvre de nouvelles politiques organisationnelles et procédures ainsi que de contrôles techniques ou physiques.

2.4 Modèles de politique de sécurité :

Nous donnerons dans cette partie un aperçu de quelques modèles de contrôle d'accès. Nous préciserons leurs limites lorsqu'ils n'arrivent pas à satisfaire les exigences en termes de protection des environnements informatiques contemporains. Ces derniers, se caractérisent par des échanges d'informations de plus en plus nombreux et importants, des accès aux informations de plus en plus complexes, des utilisateurs mobiles et des certaines opérations qui se font même à distance.

2.4.1 *Contrôle d'accès discrétionnaire DAC*

Le contrôle d'accès discrétionnaire ou DAC (Discretionary Access Control) permet de surveiller l'accès des utilisateurs aux ressources d'un système sur la base de l'identité des utilisateurs et des autorisations accordées à ces utilisateurs. Ces autorisations (ou règles) précisent, pour chaque utilisateur (ou groupe d'utilisateurs) et pour chaque ressource dans le système, les modes d'accès autorisés à ces ressources (par exemple, lire, écrire ou exécuter). Chaque demande d'un utilisateur pour accéder à une ressource (ou objet) du système est vérifiée par rapport aux autorisations spécifiées. S'il existe une autorisation indiquant que l'utilisateur peut accéder à l'objet dans le mode spécifié, l'accès est alors accordé, sinon il est refusé.

Les politiques de contrôle d'accès discrétionnaires ont, malheureusement, l'inconvénient de ne pas fournir une assurance réelle sur le flux de l'information dans un système [5] et il est souvent facile de contourner les restrictions d'accès. Par exemple, considérons trois utilisateurs : Alice, Bob et Ève. Alice crée un fichier X qui ne doit être accessible que par Bob. Le contrôle d'accès est alors spécifié comme suit :

	Fichier X	Fichier Y
Alice	Own	
Bob	Read	Write
Ève		Read

Comme Bob a le privilège Write et Ève a le privilège Read dans le fichier Y, Bob peut contourner la restriction d'accès de Ève au fichier X en copiant X dans Y. Autre exemple avec un programme cheval de Troie. Considérons la matrice de contrôle d'accès suivante :

	Fichier X	Fichier Y	Prog. Cheval de Troie
Alice	Own		
Bob	Read	Write	Execute
Ève		Read	Read, Write, Execute
Prog. Cheval de Troie	Read	Write	

Ève introduit deux opérations cachées dans l'application utilisée par Bob. Ces opérations sont Read dans le fichier X et Write dans le fichier Y. Une fois que Bob exécute l'application, les opérations Read et Write vont être permises. Puisque l'utilisateur malveillant Ève a le privilège Read dans Y, il pourra accéder à ce fichier et récupérer les informations désirées.

2.4.2 Contrôle d'accès obligatoire MAC

Le contrôle d'accès discrétionnaire est généralement défini par opposition au contrôle d'accès obligatoire ou MAC (Mandatory Access Control), qui impose des règles incontournables garantissant l'atteinte des objectifs de sécurité visés [5]. Dans ce type de contrôle d'accès les sujets ne peuvent pas intervenir dans l'attribution des droits d'accès. Ce contrôle d'accès est plus rigide que le contrôle d'accès discrétionnaire mais est, cependant, plus sûr.

Bell-LaPadula est un exemple de modèle de sécurité MAC qui a fût développé initialement pour le contrôle d'accès au sein de l'armée américaine. Bell-La Padula permet de surveiller l'accès sur la base d'une classification des utilisateurs et des ressources (objet) du système. Ainsi, à chaque utilisateur et chaque objet dans le système est associé à un niveau de sécurité. Le niveau de sécurité associé à un objet reflète la sensibilité des informations contenues dans cet objet, à savoir, les dommages potentiels qui pourraient survenir en cas de divulgation non autorisée de l'information. Le niveau de sécurité associé à un utilisateur reflète, quant à lui, la confiance que le système accorde à cet utilisateur pour ne pas divulguer des informations sensibles à des utilisateurs non autorisés à les voir. Quatre niveau de sécurité sont généralement considérés : Top Secret (TS), Secret (S), Confidentiel (C), et non classifiés (U), avec l'ordre suivant : TS > S > C > U. L'accès à un objet par un sujet n'est accordé que si deux propriétés fondamentales sont préservées : la propriété simple (no read up) et la propriété étoile (no write down). La propriété simple stipule que si nous avons un accès (sujet, objet) alors le niveau de sécurité du sujet domine (plus grand dans l'ordre) celui de l'objet. La propriété étoile stipule, quant à elle que si un sujet a accès à un objet o1 et peut altérer un objet o2 alors le niveau de sécurité de l'objet o1 doit être dominé par le niveau de sécurité l'objet o2.

Le contrôle d'accès discrétionnaire (DAC) présente de graves inconvénients vis-à-vis de ces faiblesses par rapport aux fuites d'informations, tandis que le contrôle d'accès obligatoire (MAC) est très rigide et mal adapté aux systèmes réellement répartis. Il fallait donc réaliser un compromis entre DAC et MAC et introduire le concept de rôles associés à des privilèges. Le Tableau 1 [6] donne un exemple simplifié de table 'Rôle – Privilège' qui ne considère que les opérations

d'écriture «W» et de lecture «R» renforçant la sécurité des MAC et en maintenant la flexibilité des DAC. En effet, le concept de rôle permet de fournir une classification des sujets. Et pour améliorer l'organisation, le nouveau concept nommé «groupe d'objets», permet de regrouper l'ensemble des objets passifs sur lesquels on réalise les mêmes opérations.

2.4.3 Contrôle d'accès basé sur les rôles RBAC

Une politique tels que les droits d'accès sont attribués aux utilisateurs en fonction du rôle qu'ils jouent dans le système d'information est appelée politique par rôle.

Un rôle désigne une entité intermédiaire [6] entre les utilisateurs et les privilèges. Ces derniers ne sont plus associés, d'une façon directe aux utilisateurs mais à travers des rôles. Les deux relations {Rôle, Privilège} et {Utilisateur, Rôle} définissent les privilèges accordées à chaque utilisateur.

	Dossier Administratif	Examen	Dossier Clinique	Dossier Financier	Rapport Infirmier	Prescription
Secrétaire	R W	- -	- -	R W	- -	- -
Infirmière	- -	R W	- -	- -	R W	R -
Médecin traitant	R -	R W	R W	- -	R -	R W
Pharmacien	R -	- -	- -	R W	- -	R -
Patient	R W	R -	R -	R -	R -	R -

Tableau 1 : exemple simplifié de table 'Rôle – Privilège' qui ne considère que les opérations d'écriture «W» et de lecture «R»

RBAC utilise les rôles pour gérer les privilèges. Pour cela, les utilisateurs d'un système sont associés à un ou plusieurs rôles définis à l'avance. Lorsque ceux-ci souhaitent effectuer une opération, ils vont alors activer un ou plusieurs rôles durant une session et ils peuvent effectuer toutes les opérations permises par les rôles durant cette session, grâce aux privilèges auxquels ils sont associés.

La notion de contrôle d'accès basé sur les rôles (RBAC) avait commencé avec les systèmes en ligne multi-utilisateurs et multi-applications lancés dans les années 1970. Le modèle RBAC est naturellement adapté aux organisations où les utilisateurs sont assignés aux rôles avec des privilèges de contrôle d'accès bien définis. Cependant, avec les nouvelles exigences liées aux applications dans les environnements ubiquitaire, Le modèle RBAC de base a rapidement montré ses limites et plusieurs extensions de RBAC ont été élaborés afin d'y améliorer sa sécurité. Ainsi, comme les utilisateurs sont mobiles et le nombre assez important, le contexte devient alors un facteur de première ordre dans le contrôle d'accès. Le contexte peut inclure la date, l'heure, le lieu, les capacités du système et d'autres informations représentant les entités et l'environnement. Par ailleurs, les informations du contexte présentent une dynamique élevée (variables au fil du temps). Ainsi, quelques recherches ont été menées sur la mise à jour automatique des permissions en utilisant le contexte et l'évaluation de la valeur de confiance de l'utilisateur. On présentera dans ce qui suit quelques travaux de recherche qui ont introduit des nouveaux modèles ou qui étendent le modèle RBAC, afin de répondre aux nouvelles exigences liés aux environnements informatiques ubiquitaires.

Premièrement on présentera GRBAC et DRBAC qui ont introduit la notion de contexte dans RBAC. On retiendra cette idée de prendre en considération les paramètres du contexte dans le modèle qu'on proposera plus tard. Parce que ceci va nous permettre d'avoir un modèle de contrôle d'accès dynamique, sensible au contexte et qui adapte les permissions en fonction du contexte actuel. Ensuite, on présentera les modèles qui utilisent la notion de confiance. On retiendra également cette notion pour notre modèle. Cette idée nous permettra de surmonter la faiblesse des modèles traditionnelles de contrôle d'accès où l'accès est accordé seulement sur la base d'informations d'identification de l'utilisateur. Avec la notion de confiance, nous établirons une relation de confiance avec l'utilisateur et c'est par rapport à cette relation de confiance qu'on décidera d'accorder ou de refuser l'accès. Enfin, on représentera des modèles qui ont intégré la notion de risque. Dans notre modèle, le risque sera utilisé comme un élément clé dans le processus de la prise de décision. Plusieurs techniques d'évaluation de risque ont été proposées. Pour notre modèle, on retiendra la technique d'évaluation de risque qui se base sur la notion de confiance et de contexte. De cette manière, notre modèle combinera les trois plus importantes notions qui ont récemment attiré l'attention des chercheurs dans le domaine de contrôle d'accès, à savoir, le contexte, la confiance et le risque.

2.4.4 Contrôle d'accès à base de rôle généralisé GRBAC

Moyer, M.J. et Abamad, M. ont proposé dans [7] le contrôle d'accès à base de rôles généralisés ou GRBAC (Generalized Role Based Access Control). Dans ce modèle, ils étendent le RBAC traditionnel en appliquant les rôles à toutes les entités du système. (En RBAC, le concept de rôle est utilisé uniquement pour les sujets). En définissant trois types de rôles, c'est à dire, des rôles pour les sujets, des rôles pour l'environnement, et des rôles pour les objets, GRBAC utilise les informations de contexte comme un paramètre pour prendre les décisions d'accès.

- Rôle-sujet

Un rôle-sujet dans GRBAC est analogue à un rôle du RBAC traditionnel. Chaque sujet est autorisé à assumer un ensemble de rôles-sujet. Les Rôles-sujet peuvent être hiérarchiques ou «à plat» (un seul niveau). La seule différence entre les rôles-sujet GRBAC et rôles du RBAC traditionnel est la façon dont ils sont utilisés pour prendre des décisions d'accès. En RBAC traditionnel, une décision d'accès est entièrement basée sur les autorisations associées à l'ensemble des rôles que le sujet possède. En GRBAC, par contre, une décision d'accès ne dépend pas seulement des rôles-sujet, mais aussi des rôles-environnement et des rôles-objet.

- Rôles-Environnement

Dans le monde réel, il y a beaucoup de cas dans lesquels le contrôle d'accès ne dépend pas seulement de la personne qui obtient l'accès et l'objet en cours d'accès, mais aussi de l'état de l'environnement lors de l'accès. Par exemple, de nombreuses organisations limitent l'accès à leurs systèmes pendant les nuits et les week-ends. Dans l'armée, des systèmes informatiques sécurisés sont souvent restreints au personnel dans des régions physique spécifiques, comme une salle informatique hautement sécurisé. Dans la maison, les parents peuvent restreindre l'accès de leurs enfants à la télévision, ne permettant aux enfants de regarder la télévision qu'après avoir fait leurs devoirs et jusqu'à 21h00. Dans chacun de ces cas, la politique de contrôle d'accès dépend de l'information de l'environnement. Les deux types les plus élémentaires de l'information environnementale sont le temps et le lieu, mais aussi n'importe quelle information de sécurité pertinente de l'environnement, qui peut être correctement prise en compte par le système, peut également être utilisée pour contrôler l'accès aux ressources du système.

Rôle-Objet

Rôles-sujet et rôles-environnement autorisent l'implanteur d'une politique de sécurité à structurer la politique en se basant soit sur les propriétés des sujets dans le système, ou l'état du système lui-même. Mais si l'implanteur peut aussi vouloir structurer cette dernière en fonction des propriétés des ressources dans le système. Pour tenir compte de ce scénario, le modèle GRBAC comprend également les rôles-objet. Les rôles d'objets nous permettent de capturer des points communs entre les différents objets dans un système, et utiliser ces points communs pour classer les objets dans des rôles. Les rôles des objets peuvent être basés sur n'importe quelle propriété classifiable d'un objet, y compris sa date de création, le type d'objet (image, code source, streaming vidéo, etc.), le niveau de sensibilité (secret, très secret, etc.), ou des informations sur le contenu de l'objet. Après avoir classé les objets, nous pouvons prendre des décisions de contrôle d'accès basé sur le schéma de classification ainsi créé.

2.4.5 Contrôle d'accès à base de rôle dynamique DRBAC

Dans [8], Zhang, G. et Parashar, M. utilisent également les paramètres du contexte dans leur modèle de contrôle d'accès à base de rôle dynamique DRBAC (Dynamic role-based access control model) avec deux idées principales :

(1) les privilèges d'accès d'un utilisateur doivent changer quand le contexte de l'utilisateur change.

(2) Une ressource doit ajuster son autorisation d'accès lorsque son système d'information (par exemple, la bande passante réseau, usage processeur, usage mémoire) change.

DRBAC est un mécanisme de contrôle d'accès dynamique, sensible au contexte, qui accorde et adapte de manière dynamique les autorisations aux utilisateurs en fonction du contexte actuel. Le mécanisme proposé étend le modèle (RBAC), tout en conservant ses avantages (capacité de définir et gérer des politiques complexes de sécurité). Le modèle ajuste dynamiquement les attributions des rôles (Role assignments) et les attributions des autorisations (Permission assignments) en se basant sur les informations du contexte. Dans cette approche, chaque utilisateur se voit attribuer des parties de rôles. De même, les ressources ont des sous-ensembles d'autorisation pour chaque rôle qui a accès à la ressource. L'ensemble des rôles est définis en structure hiérarchique où un rôle qui hérite d'un autre rôle se voit octroyer tout les privilèges possédés par ce dernier.

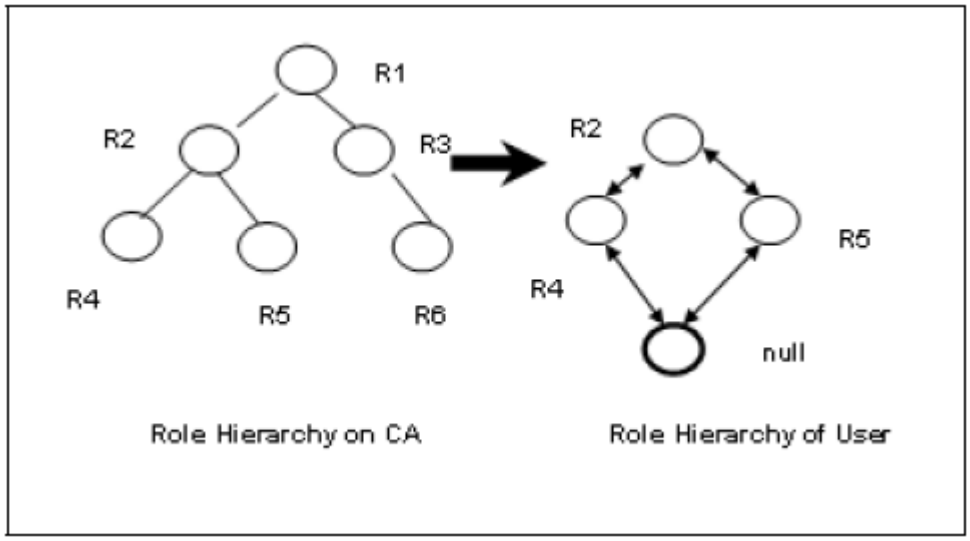


Figure 2 : Hiérarchie des rôles

La figure 2 illustre la relation entre la hiérarchie des rôles maintenue par l'Autorité Centrale (CA) et la hiérarchie des rôles assignés à un utilisateur particulier. On voit que la hiérarchie du rôle assigné à un utilisateur est un sous-ensemble de la hiérarchie globale des rôles du système.

Le fonctionnement de DRBAC est illustré en utilisant un exemple. Dans cet exemple, lorsqu'un professeur B ouvre une session du système dans son bureau avec un PDA (ordinateur de poche), l'autorité centrale (CA) lui assigne un sous-ensemble de rôles, par exemple, «Professor , «Lecturer» et «Faculty». Ensuite, l'autorité centrale met également en place un agent de contrôle d'accès sur son PDA. Un événement délivré par l'agent du contexte déclenchera alors des transitions entre les rôles dans la machine d'état des rôles. Considérons une politique de sécurité qui définit le rôle actif de B en tant que «Professor» quand il est dans le bureau (voir Figure 3, où le cercle en pointillés est le rôle actif), et définit la transition comme : changer le rôle de «Professor» à «Faculty» lorsque le professeur B quitte son bureau. Quand le professeur B accède à la ressource dans son bureau, le rôle actif «Professor» est alors utilisé. La ressource maintient des machines d'état des autorisations comme le montre la Figure 4 et où chacun des rôles, «Professor», «Faculty» et «Lecturer», ont leurs propres machines d'état des autorisations. Le cercle en pointillés représente l'autorisation active en cours pour chaque rôle. Le « null » signifie que le rôle n'est pas autorisé à accéder à la ressource.

L'agent de contexte de la ressource va déclencher des transitions dans la machine d'état des autorisations. Dans cet exemple, nous supposons que l'autorisation active pour le rôle « Professor » est P1 tandis que le chargement du système de la ressource est faible. P1 signifie à la fois lire et

écrire. La politique de sécurité de la ressource peut définir une transition d'autorisation pour le rôle «Professor» : faire transiter l'autorisation de P1 à P2 lorsque le chargement est élevé. P2 étant le privilège de lecture seule.

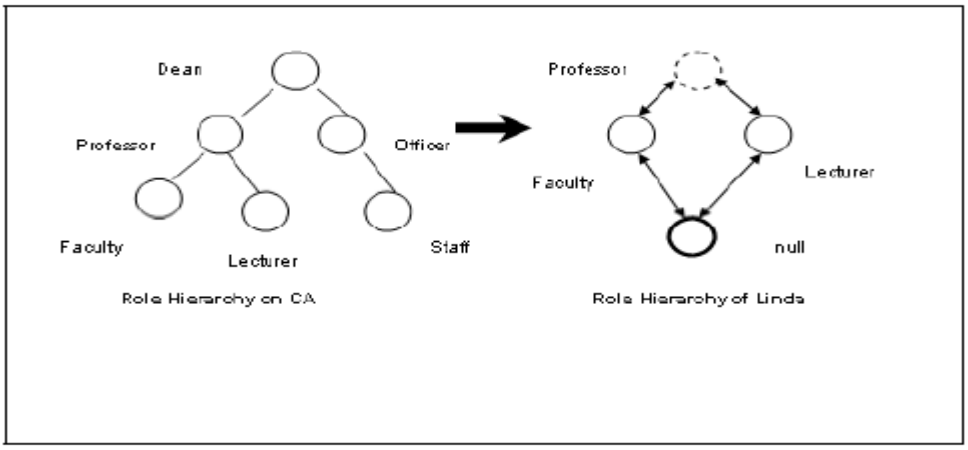


Figure 3 : Hiérarchie de rôles pour l'immeuble

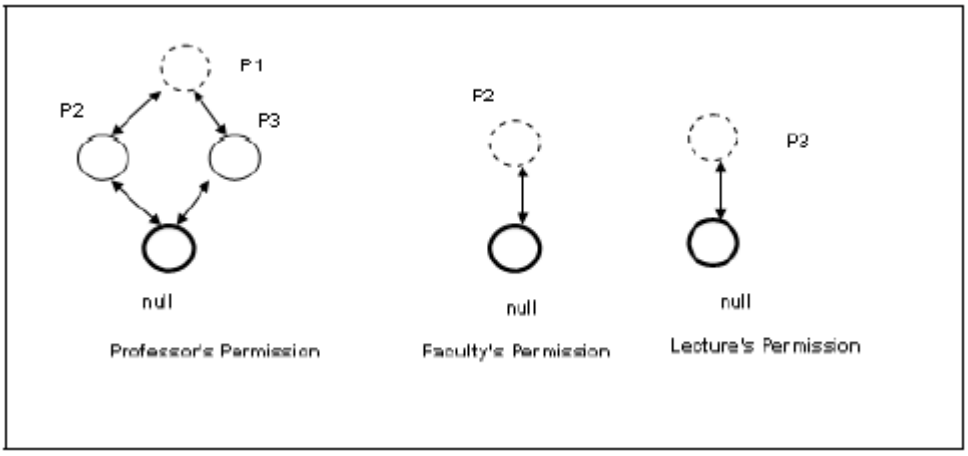


Figure 4 : Hiérarchie de permissions pour la ressource.

Sur la base des situations définies ci-dessus, nous pouvons décrire des scénarios pour illustrer le contrôle d'accès dynamique.

- ✓ Lorsque le professeur B se déplace hors de son bureau, l'agent du contexte enverra un événement à l'agent du contrôle d'accès sur son PDA. Cet événement va déclencher une transition dans la machine d'état du rôle, en changeant son rôle actif à « Faculty ». En conséquence, le professeur B ne sera pas en mesure d'écrire dans la ressource une fois qu'il sort de son bureau étant donné que le rôle « Faculty » a l'autorisation P2 ou nulle.

- ✓ Lorsque le professeur B accède à la ressource dans son bureau, son rôle actif est « Professor », qui a, à la fois, le privilège lire et écrire sur la ressource aussi longtemps que le chargement de la ressource est faible. Si le chargement devient élevé, la machine d'état de permission des ressources va changer la permission active pour le professeur B à P2 et il va perdre le privilège d'écrire dans la ressource.

Dans les scénarios décrits ci-dessus, nous voyons que DRBAC peut améliorer la sécurité des applications ubiquitaires. Le mécanisme DRBAC mis en œuvre dans cette application garantit que le privilège du professeur B pour accéder à la ressource sera modifié de manière dynamique lorsque le contexte change. L'utilisation des informations du contexte pour modifier les privilèges de l'utilisateur empêche alors que les ressources soient utilisées de façon incorrecte. Bien que GRBAC et DRBAC, rendent le contrôle d'accès plus dynamique et flexible, cependant le processus décisionnel n'est pas aussi puissant et précis que celui des modèles qui utilisent le risque. Ils ne considèrent pas l'aspect de la sécurité dans les processus de prise de décision et l'impact des problèmes de sécurité sur le système.

2.4.6 Modèle de contrôle d'accès basé sur la confiance et le contexte TCAC

Dans leur article [9], les auteurs proposent un modèle de contrôle d'accès basé sur le contexte et la confiance (TCAC) pour les systèmes ouverts et distribués.

L'attribution des rôles dans TCAC est basée sur les rôles, la confiance et le contexte du demandeur d'accès. TCAC est donc un RBAC avec la notion de confiance et de contexte. Lorsque la valeur de confiance du demandeur d'accès n'est pas inférieur au seuil de confiance prédéfini et les informations du contexte de l'utilisateur respectent les contraintes du contexte, l'utilisateur est alors affecté à certains rôles, et peut alors exécuter les autorisations associées à ces rôles. TCAC est flexible, extensible et bien adapté pour les systèmes distribués.

- ✓ <u>La confiance</u>

La confiance est une notion subjective. Cependant, la confiance peut être quantifiée en analysant le comportement des utilisateurs et nous pouvons lui attribuer une valeur dans l'intervalle réel [0, 1], 0 signifie absolument pas digne de confiance, et 1 signifie totalement fiable.

Chaque utilisateur possède une valeur de confiance particulière envers les autres à chaque instant ou au cours d'un certain intervalle de temps. La valeur de confiance d'un utilisateur change

seulement comme conséquence d'une interaction avec les autres utilisateurs. Un utilisateur avec une valeur de confiance élevée sera assigné à un rôle principal, et effectue des privilèges supérieurs, tandis qu'un utilisateur avec une valeur de confiance faible peut s'être refusé l'accès au ressources du système.

- ✓ Le contexte

Le contexte peut être utilisé pour caractériser l'environnement ou situation d'une entité. Une entité, peut être une personne, le temps ou un lieu, et est considérée pertinente pour l'interaction entre un utilisateur et une application. Nous pouvons définir des contraintes contextuelles en fonction des différents besoins de l'application. Selon les informations du contexte courant des utilisateurs, le système de contrôle d'accès décide alors si les contraintes contextuelles sont satisfaites ou non et donne l'accès aux ressources en conséquence.

Les composants fondamentaux de TCAC sont illustrés dans la Figure 5, avec U, T, SUB, R, OPS, OBJ, PRMS, Context et SES représentent, respectivement, l'ensemble des utilisateurs, les valeurs de confiance, les sujets, les rôles, les opérations, les objets, les permissions, le contexte et les sessions. RA est la relation d'affectation des sujets-aux-rôle et PA est la relation d'affectation des autorisations-aux-rôle.

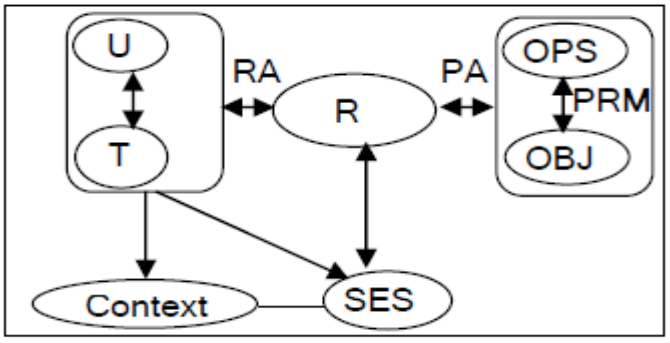


Figure 5 : Le modèle TCAC

La politique de l'accès avec TCAC est comme suit. Une demande d'accès est un triplet $\prec s, p, o \succ \in SES \times OPS \times OBJ$, qui signifie qu'un utilisateur, dans une session s, veut effectuer une certaine opération p sur l'objet o. Lorsqu'un utilisateur envoie une demande d'accès, ses informations courantes, y compris son identité, sa valeur de confiance et l'information contextuelle peut être obtenue par le système. La politique d'accès du TCAC stipule que si la valeur de la confiance de l'utilisateur n'est pas inférieur au seuil de la confiance du système et les informations

du contexte satisfont les contraintes du contexte, l'utilisateur est assigné à certains rôles, et pourra bénéficier des autorisations correspondantes.

Dans les systèmes distribués, il existe un grand nombre d'utilisateurs, seuls les utilisateurs de confiance peuvent accéder aux ressources et les utilisateurs qui ne sont pas dignes de confiance seront alors refusés. TCAC introduit le mécanisme de réputation, basé sur l'évaluation de la confiance. Les mécanismes de la réputation, basés sur l'évaluation de la confiance, peuvent être classés en deux types : mécanismes centralisés et distribués.

Les mécanismes centralisés sont généralement utilisés dans certains systèmes de commerce électronique, comme eBay et Alibaba. Les utilisateurs de ce genre de systèmes sont considérés comme des nœuds, et une interaction entre deux nœuds est appelée une transaction. Les deux parties donnent des évaluations sur la transaction en cours, tels que 1, 0 ou -1, indiquant respectivement bon, moyen ou mauvais. Les informations sur l'utilisateur et le rapport de la transaction sont gérés de manière centralisée. Ceci est simple et efficace, mais la prémisse est que les utilisateurs devront avoir suffisamment de confiance au système.

L'idée de base des mécanismes distribués est que chaque nœud dans le réseau calcule les valeurs de la réputation locale des autres nœuds selon les informations locales, y compris l'historique des transactions et les relations de confiance, et obtient la matrice de confiance initiale, puis obtient les valeurs de la réputation globale en étendant le champ de confiance des nœuds.

- ✓ Réputation et Confiance

Il faut d'abord préciser la différence entre la réputation et la confiance. En effet, la réputation est un concept important pour le mécanisme d'évaluation de la confiance, mais elle est toujours associée et confondue avec la confiance, il est donc important de faire la différence entre la réputation et la confiance. La confiance est en général un critère subjectif, et la confiance envers un utilisateur peut être vue comme une prédiction sur l'action future de cet utilisateur. Un facteur important affectant la confiance est la réputation de l'utilisateur. Cependant la réputation est passive. Un utilisateur ne peut pas décider de sa réputation, mais il peut affecter sa réputation en agissant honnêtement ou dans l'autre sens. La réputation d'un utilisateur est évaluée du point de vue des autres utilisateurs et est souvent basée sur ses expériences et les observations de ses actions passées.

Même si la confiance et la réputation sont des concepts différents, ils sont tous les deux dépendants du contexte et sont, par excellence, des paramètres dynamiques. Dans différents environnements contextuels et dans le temps, ils sont évalués sur la base du comportement récent et passé (historique). La valeur de confiance d'un utilisateur peut être évaluée par sa valeur de la réputation. Si la valeur de la réputation locale d'un utilisateur est « t » et la valeur de la réputation globale est « T », alors la valeur de la confiance est souvent calculée comme suit :

$$Trust = \alpha * t + (1- \alpha) * T \qquad 0 < \alpha < 1,$$

où α est le coefficient de poids défini par le système selon les applications.

L'évaluation de la réputation locale et globale est faite de la même manière que celle représentée dans le modèle RBAC dynamique avec évaluation de la confiance pour les systèmes multi-agents. La figure 6 présente la mise en œuvre proposée du système de contrôle d'accès basé sur la confiance et le contexte TCAC.

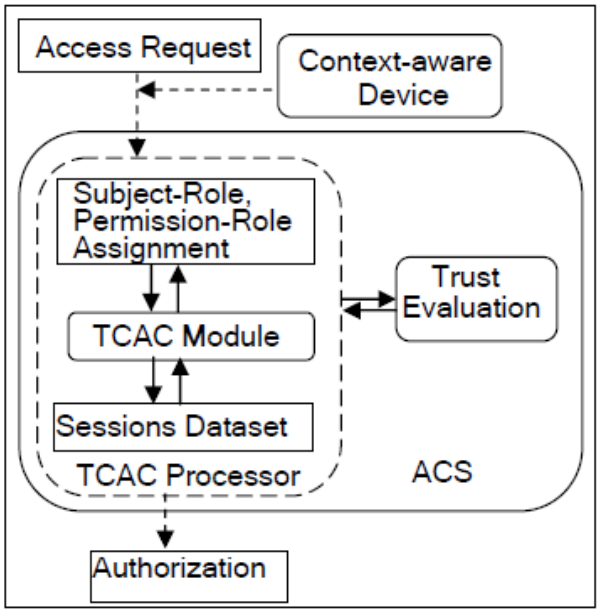


Figure 6 : Implémentation de TCAC

Cette mise en œuvre comprend deux composantes:

1. Context-Aware Device : reçoit les informations du contexte du demandeur d'accès et les envoie au système de contrôle d'accès.

2. Access Control System (ACS) : permet d'empêcher la divulgation des informations privées des utilisateurs, l'ACS est la seule composante à laquelle il est permis d'interroger le Contexte-Aware Device. ACS comprend deux parties:
    - Le processeur TCAC (Processor TCAC) : décide d'accepter ou non les demandes d'accès en fonction des informations de l'utilisateur, des informations du contexte et des politiques d'accès, et envoie les informations sur le comportement de l'utilisateur pendant la session au module de l'évaluation de la confiance (Trust Evaluation) ;
    - Trust Evaluation : calcule la valeur de confiance du demandeur d'accès en fonction de son comportement après avoir fini ses opérations, et met à jour les rapports.

Le processus d'une demande d'accès est illustré à la Figure 7. On suppose que dans une session « s » un utilisateur « u » avec une valeur de confiance « t_u » veut effectuer une opération « p » sur l'objet « o ». La politique de contrôle d'accès est comme suit : si la valeur de confiance de l'utilisateur n'est pas inférieure à 0,5 et les informations de contexte y compris le temps de la demande d'accès et l'emplacement satisfont CC = {heure ∈ [9 AM, 17 PM], loc = Office}, la demande sera accordée, l'utilisateur sera affecté au rôle R et peut alors effectuer les autorisations correspondantes.

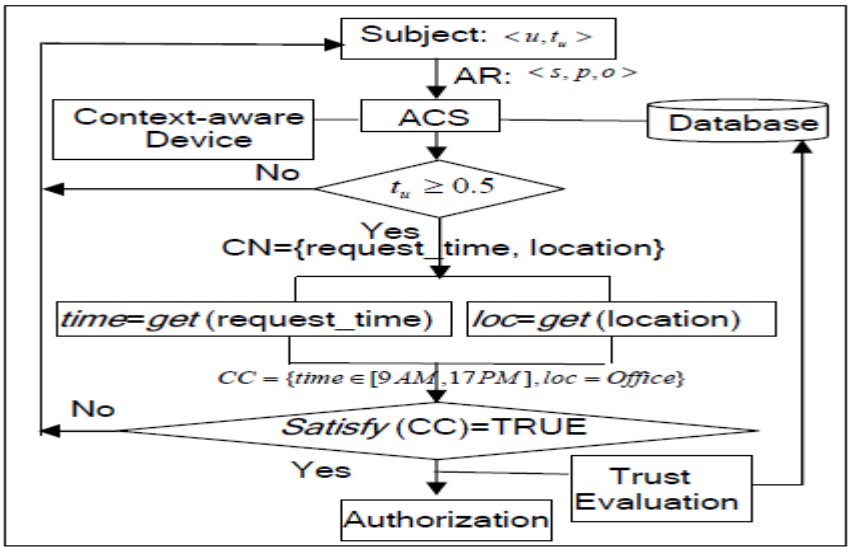


Figure 7 : Le processus d'une demande d'accès

Le processus du contrôle d'accès est détaillé comme suit :

1. Dans une session « s », un utilisateur « u » avec une valeur de confiance « t_u » envoie une demande d'accès (AR) <s, p, o> à l'ACS, ce qui signifie qu'il veut effectuer l'opération « p » sur « o »;
2. L'ACS obtient les informations de l'utilisateur telles que son identité et sa valeur de confiance, et décide si « t_u » n'est pas inférieure à 0,5. Si c'est le cas, le rôle « r » devient alors le rôle de la session et exécute l'étape 3, sinon, l'ACS rejette la requête ;
3. L'ACS obtient l'information du contexte courant, y compris l'heure de la demande et l'emplacement, par la fonction « get » à travers le context-aware device, et décide si les contraintes du contexte CC = {heure ∈ [9AM, 17PM], loc = Office} sont satisfaites. Si c'est le cas, le rôle r devient alors actif et exécute l'étape 4, sinon, il refuse la demande;
4. L'ACS accorde la demande d'accès, et assigne l'utilisateur « u » au rôle « r », en conséquence, l'utilisateur peut effectuer la permission (p, o). Enfin, la valeur de confiance du demandeur d'accès est évaluée par le mécanisme d'évaluation de confiance « Trust Evaluation » en fonction de son comportement pendant la session, et mise à jour dans la base de données locale.

Le modèle TCAC pour les systèmes distribués étend le RBAC en introduisant la notion de la confiance et du contexte. L'acceptation d'une demande d'accès est déterminée par deux facteurs : la valeur de confiance de l'utilisateur qui ne doit pas être inférieure au seuil de confiance défini dans les politiques de sécurité, et les contraintes du contexte qui doivent être satisfaites. TCAC peut attribuer dynamiquement les rôles en fonction du comportement des utilisateurs et la situation du contexte au lieu de se limiter à vérifier l'identité, ce qui rend ce modèle plus souple, évolutif et bien adapté aux systèmes dynamiques et distribués.

Dans la prochaine section nous présentons une autre extension du modèle RBAC avec la notion de confiance pour les systèmes multi-agents.

2.4.7 *Modèle RBAC dynamique avec évaluation de la confiance pour les systèmes multi-agents*

Dans [10], les auteurs proposent une nouvelle approche de contrôle d'accès pour les systèmes multi-agents basée sur le modèle (RBAC). Afin de mettre en évidence les changements dynamiques

de l'environnement et des rôles affectés à un utilisateur, le schéma proposé emploie la notion de confiance évaluée par la mesure de la satisfaction et de la réputation. Il actualise également le rôle de l'utilisateur et les autorisations selon les informations du contexte. Dans ce travail, les auteurs proposent une méthode d'évaluation de la confiance en utilisant le niveau de la satisfaction et de la réputation. La réputation est l'évaluation de la confiance de l'agent demandeur «requesting agent» et la satisfaction est l'évaluation de la confiance de l'agent demandé «requested agent». La valeur de la confiance d'un agent peut être évaluée par son degré de satisfaction et par la valeur de sa réputation. Aussi, les auteurs ont appliqué un ensemble de règles afin de mettre à jour, dynamiquement, le rôle de l'utilisateur selon les informations du contexte.

En système multi-agents, chaque composant peut être considéré comme un agent et la structure du modèle proposé est résumée à la Figure 8.

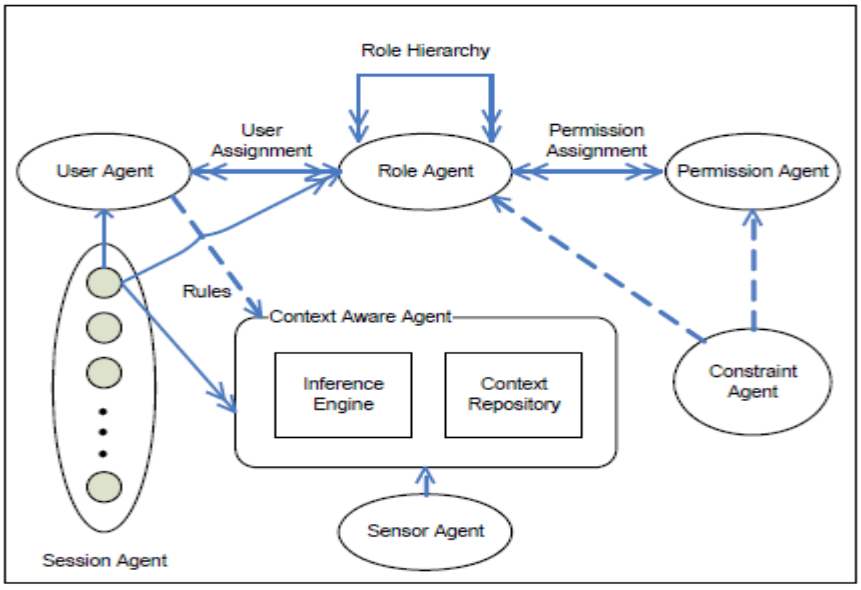


Figure 8 : Le modèle RBAC dynamique pour les systèmes multi-agents

Ici, 'User Agent' est le même que «utilisateur» du modèle original RBAC. «Role Agent" tient la liste des rôles et gère la hiérarchie des rôles. «Permission Agent» tient la liste des autorisations. Cependant, contrairement à RBAC originale, «Session Agent» enregistre les règles dans «Context-Aware Agent» avant la connexion de «User Agent» et «Permission Agent». Plus d'une règle peuvent être assignée à chaque session dans «Session Agent». Ces règles sont tirées à l'aide des informations sur l'utilisateur et du contexte obtenues à partir de divers «Sensor Agent». Quand une règle enregistrée est tirée, «Session Agent» met à jour, dynamiquement, le rôle de l'utilisateur en fonction du contexte. «Constraint Agent» vérifie les contraintes lorsque les rôles et les autorisations

sont attribués. Il vérifie aussi la valeur de confiance qui est évaluée par la valeur de la satisfaction et de la réputation. Si elle est acceptable, «Constraint Agent» permet alors l'autorisation autrement il la bloque. «Context-Aware Agent», qui est composé de «Inference Engine» et «Context Repositery», conclut le contexte en utilisant les divers informations du contexte et indique le résultat lorsque la règle est déclenchée. «Sensors Agents» recueille les informations du contexte et les envoie à «Context-Aware Agent».

- ✓ Confiance de la Satisfaction et la Réputation

Dans le modèle précédent TCAC, la confiance est évaluée en utilisant une réputation locale et globale. Toutefois, cette information n'est pas suffisante pour évaluer précisément la confiance d'un agent. Pour calculer la valeur de confiance, les auteurs introduisent la notion de satisfaction qui représente la confiance aux services et aux ressources que les agents fournissent. On considère d'abord le facteur clé de la proposition du modèle de la confiance-satisfaction et réputation (T-SR), qui est l'évaluation de la confiance. Nous introduisons ensuite les définitions et les méthodes nécessaires pour gérer la confiance.

- ✓ L'évaluation de la confiance

Un agent de système multi-agent peut être un utilisateur ou un certain type de dispositif qui peut fournir et demander des ressources ou des services. Si l'un des agents veut demander un service, d'abord, il doit passer le test de l'autorisation pour vérifier s'il est autorisé ou non. Ensuite, il faut résoudre le problème de la sécurité. Même si l'agent est autorisé, il y a toujours la possibilité que le service soit abusé. Comme il n'existe pas d'autorité de certification centrale dans le système multi-agents, un nouvel agent doit vérifier la valeur de la confiance des agents en utilisant l'information de la satisfaction et de la réputation.

Contrairement au modèle TCAC, on propose le modèle de satisfaction pour évaluer la confiance dans les deux sens.

La figure 9 montre la structure de l'évaluation de la confiance. Lorsque «A» demande une ressource de «B», «B» vérifie si «A» fait cela avec un but honnête, la mesure de la réputation. En même temps, «A» vérifie si «B» fournit correctement la ressource, la mesure de la satisfaction.

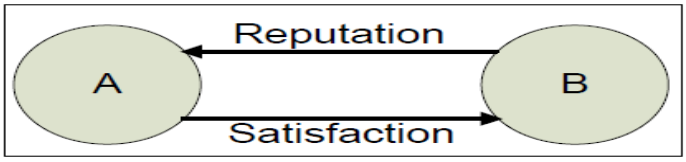


Figure 9 : La structure de l'évaluation de la confiance

La valeur de la confiance d'un agent peut être évaluée par son degré de satisfaction et la valeur de sa réputation. Si le degré de satisfaction est notée SD et la valeur de la réputation est notée R, on calcule la valeur de confiance comme suit :

$$Trust = \varepsilon_1 * SD + \varepsilon_2 * R,$$

$$\varepsilon_1 + \varepsilon_2 = 1, \varepsilon_1 \succ 0, \varepsilon_2 \succ 0,$$

Où ε_1 et ε_2 sont les coefficients de poids définis par le système selon les applications.

- ✓ <u>La satisfaction</u>

Afin de calculer le SD_i, qui représente le degré de satisfaction de l'agent-i, nous devons savoir S_{ji} qui est le niveau de satisfaction évalué par l'agent-j.

$$SD_i = \frac{\sum_{j \in V_i} S_{ji}^{W_j}}{n}, 0 \le S \le 1,$$

où n est le nombre des agents qui ont demandé des services, et V_i est l'ensemble de ces agents. Avec cette équation, nous pouvons obtenir les propriétés suivantes:

- SD_i est comprise entre 0 et 1. Si elle est proche de 0, cela signifie que l'agent-i n'est pas digne de confiance, et si elle est proche de 1, l'agent-i est digne de confiance.
- Le coefficient de poids, W_j, montre l'importance de l'avis de l'agent-j.

- ✓ <u>La réputation</u>

Les réputations sont évaluées en fonction du comportement récent et de l'historique du passé. Un agent évalue la réputation locale d'un autre agent en fonction de l'expérience d'interaction avec lui, et l'historique des interactions peut servir comme référence pour l'évaluation de la réputation.

Soit r_{ij} la valeur de la réputation locale de l'agent-j évaluée par l'agent-i, qui peut être calculé comme suit :

$$r_{ij} = \frac{N_h(i,j)}{N_h(i,j) + N_m(i,j) * Pnsh}$$

Où $N_h(i,j)$ et $N_m(i,j)$ désignent, respectivement, le nombre de transactions honnêtes et malicieuses entre l'agent-i et l'agent-j; Pnsh désigne le coefficient de châtiment des opérations malveillantes, évidemment Pnsh > 1. Pnsh est le coefficient défini par le système pour régler la sensibilité du niveau de sécurité. Selon l'équation précédente, plus la transaction est honnête, plus la valeur de la réputation est proche de 1. Sinon, la valeur se rapproche de 0.

La réputation globale d'un agent est évaluée par l'ensemble du système multi-agents. Soit R la valeur de la réputation globale de l'agent-i. La méthode d'obtention de la valeur de réputation globale est de calculer la moyenne des valeurs de réputation locale d'un agent évalué par d'autres agents. En utilisant les r valeurs, la valeur R peut être calculée comme suit.

$$R_i = \frac{\sum_{j \in V_i} R_j^{N_w} * (1 - e^{-\frac{N(j,i)}{5}}) * r_{ji}}{\sum_{j \in V_i} R_j^{N_w} * (1 - e^{-\frac{N(j,i)}{5}})}$$

Où N_w dénote l'importance des avis des agents qui ont des valeurs de confiance élevées.

Les opinions des différents agents sont traitées différemment de telle sorte que les opinions des agents de valeurs de confiance élevées sont jugés plus importantes que ceux de valeurs de confiance faibles.

$N(j,i) = N_h(j,i) + N_m(j,i)$ représente le nombre total des transactions. La réputation globale est accumulée progressivement. Plus les transactions honnêtes continuent d'exister, plus la valeur de réputation globale est forte. Pour cette raison, la fonction de poids, $1 - e^{-\frac{N(j,i)}{5}}$, est mise. Il s'agit

d'une fonction convexe, qui peut efficacement réguler l'importance des transactions. Enfin, nous avons l'équation suivante qui nous donne la valeur la confiance d'un agent.

$$Trust_i = \varepsilon_1 * \frac{\sum_{j \in V_i} S_{ji}^{w_j}}{n} + \varepsilon_2 * \frac{\sum_{j \in V_i} R_j^{N_{wj}} * (1 - e^{-\frac{N(j,i)}{5}}) * \frac{N_h(i,j)}{N_h(i,j) + N_m(i,j) * Pnsh}}{\sum_{j \in V_i} R_j^{N_{wj}} * (1 - e^{-\frac{N(j,i)}{5}})}$$

- ✓ <u>Appliquons le modèle T-SR:</u>

Le processus d'autorisation d'accès est illustré dans la Figure 10.

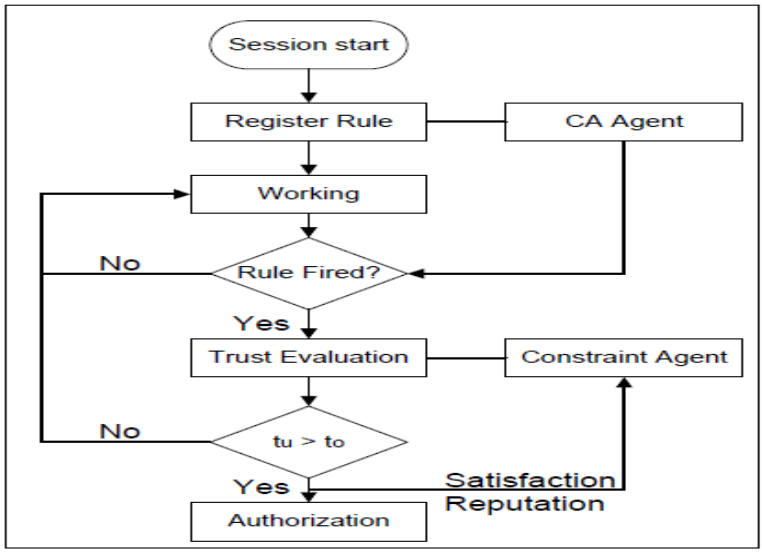


Figure 10 : Le processus de l'autorisation d'accès

Le processus d'autorisation d'accès est résumé ci-dessous:

- Étape 1: Après qu'une session est lancée, la règle pour l'utilisateur est enregistrée dans le «CA Agent».
- Etape 2: Pendant que l'utilisateur travaille, le «CA Agent» continue à surveiller les informations du contexte.
- Étape 3: Si la règle est tirée, la confiance est évaluée par le «Constraint Agent». Sinon, l'utilisateur continue à travailler.

- Étape 4: Si la valeur de confiance de l'utilisateur, t_u, est supérieur au seuil t_0, l'autorisation est accordée. Sinon, la mise à jour du rôle est refusée. Entre-temps, la valeur de satisfaction et la valeur de la réputation sont mises à jour.

Le modèle T_SR est une approche dynamique de RBAC employant la confiance de satisfaction et de réputation pour les systèmes multi-agents. Il permet l'affectation dynamique du rôle, ce qui est important pour un système multi-agent dans un environnement d'informatique ubiquitaire. Les lacunes de l'affectation statique des rôles dans le modèle RBAC traditionnel et la question de sécurité ont été résolues. Le modèle RBAC existant emploie une approche unidirectionnelle d'évaluation de la confiance qui rend difficile d'atteindre une valeur de confiance précise. Le modèle T-SR proposé, peut évaluer non seulement la valeur de la confiance de l'agent qui demande un service ou une ressource, mais aussi celle de l'agent qui fournit le service ou la ressource. Par conséquent, il permet d'avoir une valeur de confiance d'une grande précision.

Nathan Dimmock dans son article [11] définit le contrôle d'accès basé sur la confiance comme suit :

«Trust based access control is the idea of using a model of human notions of trust and community as the basis for assigning privileges.».

Cependant, bien que nous puissions faire des déclarations générales telles que «j'ai confiance en Brian », ou même des déclarations quantitatives telles que «J'ai confiance de 90% en Brian», ce ne sont pas particulièrement significatives.

Quelle confiance ai-je besoin d'avoir en mon collègue avant, par exemple, de lui emprunter mon téléphone?

Dans les communautés humaines, le montant de la confiance nécessaire semble dépendre de la nature de l'action, ou plus précisément, du risque encouru. La probabilité qu'une personne prêtera un montant donné d'argent à un ami dépend du montant en cause, la qualité de l'ami, et peut-être d'autres facteurs tels que les expériences passées de prêt d'argent à cette personne. Effectivement, la personne se pose la question suivante : Quelle est la probabilité que cette personne me rembourse et, si elle ne le fait pas, qu'est-ce que cela peut générer?

Intégrer l'évaluation des risques dans les systèmes de contrôle d'accès a récemment attiré l'attention des chercheurs. Un bref aperçu de quelques-uns des travaux existants est donnée ci-dessous.

2.4.8 Contrôle d'accès avec évaluation de risque

L'article [12] introduit un nouveau modèle 'Acces Control Model with risk assessment' d'évaluation des risques dans un environnement ubiquitaire et utilise le risque comme un élément clé dans le processus de prise de décision. Cette solution permet une gestion de contrôle d'accès plus dynamique et plus précise.

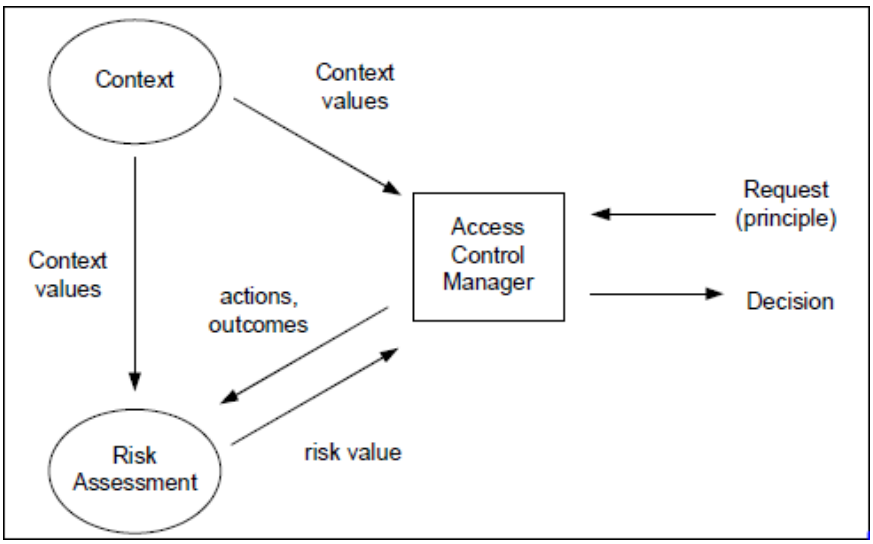


Figure 11 : La structure du contrôle d'accès

Il y a trois modules dans le système comme le montre la figure 11. Dans ce système, le gestionnaire de contrôle d'accès «Access Control Manager » est le module principal. Il reçoit les demandes d'accès, les analyse, collecte d'autres paramètres et envoie les données au module d'évaluation des risques «Risk Assessment». Après cela, il prend des décisions pour chaque demande en fonction de la valeur du risque fournie par le module d'évaluation des risques.

L'évaluation des risques «Risk Assessment» est un module qui joue un rôle clé dans la structure. Il calcule les valeurs des risques en se basant sur les données d'entrée du gestionnaire de contrôle d'accès et des données du contexte à partir du module «Context».

Le module contexte «Context» a la responsabilité de collecter des paramètres des utilisateurs et de l'environnement pour supporter les autres modules. Dans cet article, les auteurs ne

mentionnent pas comment collecter les données du contexte pour les utilisateurs et l'environnement.

Une demande de l'utilisateur p d'exécuter une action est soumise au gestionnaire de contrôle d'accès. Le gestionnaire de contrôle d'accès regarde les «outcomes» pertinents qui peuvent se produire en raison de cette action et interroge le module d'évaluation des risques pour le calcul de la valeur du risque après lui avoir adressé les paramètres nécessaires. Le module d'évaluation des risques, après le calcul du coût des «outcomes» en termes de disponibilité, de confidentialité, et d'intégrité et en fonction du contexte de l'utilisateur, de l'environnement et des ressources, évalue la valeur du risque de l'action. La décision est prise par le gestionnaire de contrôle d'accès en se basant sur la valeur du risque fournie par le module d'évaluation des risques. La valeur du risque est comparée avec un seuil, puis le gestionnaire du contrôle d'accès donne la décision.

2.4.9 Context-Risk-Aware Access Control (CRAAC)

Le modèle CRAAC a été introduit dans l'article [13], il est bâti sur la base du RBAC.

Les ressources/les services sont classées dans des groupes d'objets ayant chacun un niveau de confiance minimum pour autoriser l'accès (OLoA : Object Level of Assurance).

L'OloA est déterminé en se basant sur le niveau de sensibilité et l'impact d'un accès non autorisé à cet objet.

Au moment de l'exécution, les informations du contexte sont évaluées pour déterminer le niveau de confiance du demandeur de l'accès (RLoA : Requester's level of Assurance).

L'accès n'est autorisé que lorsque RLoA >= OLoA

Comment déterminer le RLoA?

La détermination du RLoA se fait en temps réel, donc il est nécessaire de fixer un ensemble d'attributs du contexte qui ont un impact sur le niveau du risque d'un accès non autorisé.

En effet, les facteurs qui déterminent le risque d'un accès non autorisé sont : les protocoles faibles d'authentification/authentifieur(token), les emplacements d'accès qui ne sont pas dignes de confiance, les canaux de communication non protégés, le manque de fiabilité des systèmes de détection d'intrusions, etc.

Dans l'article, les auteurs se concentrent sur quatre attributs :

- Les authentifeurs (Authentification token types)
- L'emplacement d'accès (Access Locations)
- La sécurité du canal de communication (Channel Security)
- La réaction aux intrusions (Intrusion Response)

✓ Les authentifeurs (Authetification token types)

Plusieurs facteurs dans l'authentification électronique affectent le niveau de confiance pour vérifier l'identité du demandeur d'accès, dans l'article on se concentre sur le type des authentifieurs (eTokens).

Les authentifieurs ont des différents degrés de confiance pour l'identification et l'authentification. Pour quantifier le degré de confiance, on introduit le LoA(eToken).

LoA(eToken) est le degré de confiance qu'un authentifieur présente la véritable identité de l'utilisateur.

Les types des eTokens et le niveau de confiance de chaque type, selon NIST(National Institute of standards and technology), sont comme suit :

Token Type	Levels			
	1	2	3	4
Hard Token	✓	✓	✓	✓
One-time password token	✓	✓	✓	
Soft token	✓	✓	✓	
Password token	✓	✓		

Table 2 : Les types des eTokens et leurs LoA(eToken)

- ✓ L'emplacement d'accès (Access Locations)

Il existe deux types d'identification :

- Identification électronique : l'utilisateur s'identifie à travers un authentifieur (eToken)
- Identification physique «p-authentification» : l'utilisateur s'identifie à travers l'identification par la biométrie, des capteurs d'identification ou des services d'identification basés sur l'emplacement.

L'emplacement d'accès est un attribut de contexte important dans un environnement d'informatique ubiquitaire. En plus de l'attribut d'authentification eToken, l'article introduit un autre attribut d'authentification «Access Location(ALoc)».

Aloc est le degré de confiance en un emplacement d'accès.

La méthode de représentation de zones qui a été introduite dans les articles intitulés «Activity zones for context-aware computing» et «A flexible location-context representation» a été utilisé dans cet article afin de définir des différentes zones et un niveau de confiance pour chacune (LoA(ALoc)) comme le montre le tableau suivant :

Alternatives	LoA(ALoc)
Zone-0	Niveau 0; des zones publiques qui ne présentent aucun service pour l'identification physique.
Zone-1	Niveau 1; zone semi-publique qui utilisent l'identification physique pour identifier un groupe d'utilisateurs.
Zone-2	Niveau 2; zone personnelle : l'accès à cette zone est contrôlé par l'utilisation d'une clé de casier ou par un capteur d'identification.

Zone-3	Niveau3; zone personnelle sécurisée : dans cette zone on utilise certaines formes de méthodes fortes d'identification qui sont moins vulnérables au vol ou à la perte des clés de casier : l'identification biométrique par exemple.
Zone-4	Niveau4; zone personnelle très sécurisée : cette zone utilise plusieurs méthodes d'identification physique.

Table 3 : Les alternatives des emplacements et leurs LoA(ALoc)

Les attributs eToken et Aloc, ensemble, portent une contribution dans le niveau de confiance de l'identification d'un utilisateur, on introduit alors la notion de LoA(authN).

LoA(authN) est le niveau de confiance associé à l'identification par l'identification électronique (eToken) et l'identification physique(p-authentication).

- ✓ La sécurité du canal de communication (Channel Security)

Le niveau de confiance de la sécurité du canal de communication influence le niveau du risque d'un accès non autorisé, on introduit alors l'attribut du contexte CS (sécurité du canal).

LoA (CS) est le niveau de confiance en la sécurité du canal qui relie le demandeur d'accès au fournisseur du service.

LoA (CS) et le LoA (ALoc), contrairement au LoA (eToken), n'ont pas un consensus international pour déterminer les différents niveaux de confiance. Le tableau suivant est juste un exemple:

LoA(CS)	Descriptions
Niveau 0	L'attribut LoA(CS) n'est pas utilisé.
Niveau 1	Pas de confiance(ou faible confiance) à la sécurité du canal.
Niveau 2	Certaine confiance à la sécurité du canal.

Niveau 3	Grande confiance à la sécurité du canal.
Niveau 4	Très grande confiance à la sécurité du canal.

Table 4 : LoA(CS)

- ✓ La réaction aux intrusions (Intrusion Response)

Le dernier attribut du contexte est la réaction aux intrusions (IR).

LoA (IR) est le degré de confiance en un système de pouvoir détecter les intrusions et répondre à certaines attaques.

Selon [14], la capacité de détecter une intrusion et d'y répondre immédiatement varie d'un système de détection à un autre. Donc, comme les capacités de détection sont différentes, on peut associer des niveaux de confiance à chaque classe des systèmes de détection d'intrusions.

Le tableau suivant ne représente qu'un exemple donné par les auteurs car il n'existe pas encore un consensus international pour déterminer les différents niveaux de confiance à chaque classe des systèmes de détection d'intrusions.

LoA(IR)	Descriptions
Niveau 0	Aucun système de détection d'intrusions n'est installé.
Niveau 1	Pas de confiance (ou faible confiance) au système de détection d'intrusions installé.
Niveau 2	Certaine confiance au système de détection d'intrusions installé.
Niveau 3	Grande confiance au système de détection d'intrusions installé.
Niveau 4	Très grande confiance au système de détection d'intrusions installé.

Table 5 : LoA(IR)

Après que tous les attributs du contexte sont identifiés ainsi que leurs niveaux de confiance, on passe à déterminer la valeur totale qui est la valeur du niveau de confiance (LoA) d'un demandeur d'accès en se basant sur le niveau de confiance des attributs.

Le niveau de la confiance pour identifier un utilisateur est influencé par un ensemble d'attributs du contexte : directement (eToken, l'emplacement d'accès), ou indirectement (le canal de communication, les réactions aux intrusions).

Pour quantifier la valeur de confiance d'un demandeur d'accès on introduit le RLoA qui est la combinaison des valeurs de confiance des différents attributs du contexte (eToken, ALoc, CS, IR).

Pour déterminer le RLoA, les LoA des attributs (LoA (eToken), LoA (ALoC), LoA (CS), LoA (IR)) doivent être convertis de niveaux à des taux (poids).

La conversion est faite par la méthode ROCs (Rank Order Centroids), qui a été proposée dans [15].

Après la conversion, on commence par calculer le LoA des deux attributs eToken et ALoc : LoA(authN) = 1 – (1- LoA(eToken)) (1-LoA(Aloc))

Et le RLoA (Requester's level of Assurance) sera calculé comme suit :

RLoA = min (LoA(authN,LoA(IR), LoA(CS))

Où min est la fonction qui retourne la valeur minimale entre ces paramètres.

Dans cet article, les auteurs ont introduit un nouveau modèle de control d'accès, CRAAC, pour les environnements d'informatique ubiquitaire. L'évaluation des risques et le niveau de confiance jouent un rôle clé dans ce modèle. Les ressources sont classées en différents groupes selon leurs niveaux de sensibilité et les impacts d'un accès non autorisé. On attribue à chaque groupe un OloA distinctif indiquant le niveau de confiance minimal pour autoriser l'accès à ce groupe de ressources. Lors de la réception d'une demande d'accès aux objets, le demandeur de l'information est évalué, et un RLoA est déterminé à partir de ses informations contextuelles. La demande d'accès est accordée si et seulement si RLoA ≥ OloA.

Dans ces deux travaux (le contrôle d'accès avec évaluation de risque et le CRAAC), les auteurs n'ont pas intégré le comportement passé des utilisateurs afin d'évaluer les risques.

Ce défaut est résolu avec l'approche suivante.

2.4.10 Risk-based Decision Method for Access Control System

L'article [16] propose une méthode de décision basée sur le risque pour les systèmes de contrôle d'accès. Une des nouveautés de ce travail est de calculer dynamiquement les valeurs de la confiance et du risque pour chaque paire sujet-objet. La décision est prise en se basant sur ces valeurs. Par ailleurs, les deux valeurs sont adaptatives, reflétant le comportement passé des utilisateurs avec des objets particuliers.

Dans ce travail, on évalue le comportement du passé en se basant sur l'historique des récompenses et des pénalités. À la fin de chaque opération, des points de récompense ou de pénalités sont attribués à l'utilisateur par rapport à des objets spécifiques.

Le flux de ce processus est montré dans la figure 12. Cette structure est une modification de la structure de la norme XACML. Tous les nouveaux composants qui ont été ajoutés sont encadrés par des lignes pointillées.

Traditionnellement, chaque fois qu'un point de décision de politique PDP (Policy Decision Point) reçoit une demande d'accès du demandeur, il demande d'abord des informations supplémentaires du point d'administration des politiques PAP (Policy Access Point) et du point d'information de politique PIP (Policy Information Point), puis il prend une décision. Dans la méthode proposée, le PDP demande également des informations au sujet des valeurs de la confiance et du risque associées à un objet et un sujet particulier, puis il prend la décision.

Ces valeurs de confiance et de risque sont calculées en fonction de trois facteurs principaux:

1) l'historique des récompenses et des pénalités,

2) Niveau d'autorisation du sujet, et

3) Niveau de sensibilité de l'objet.

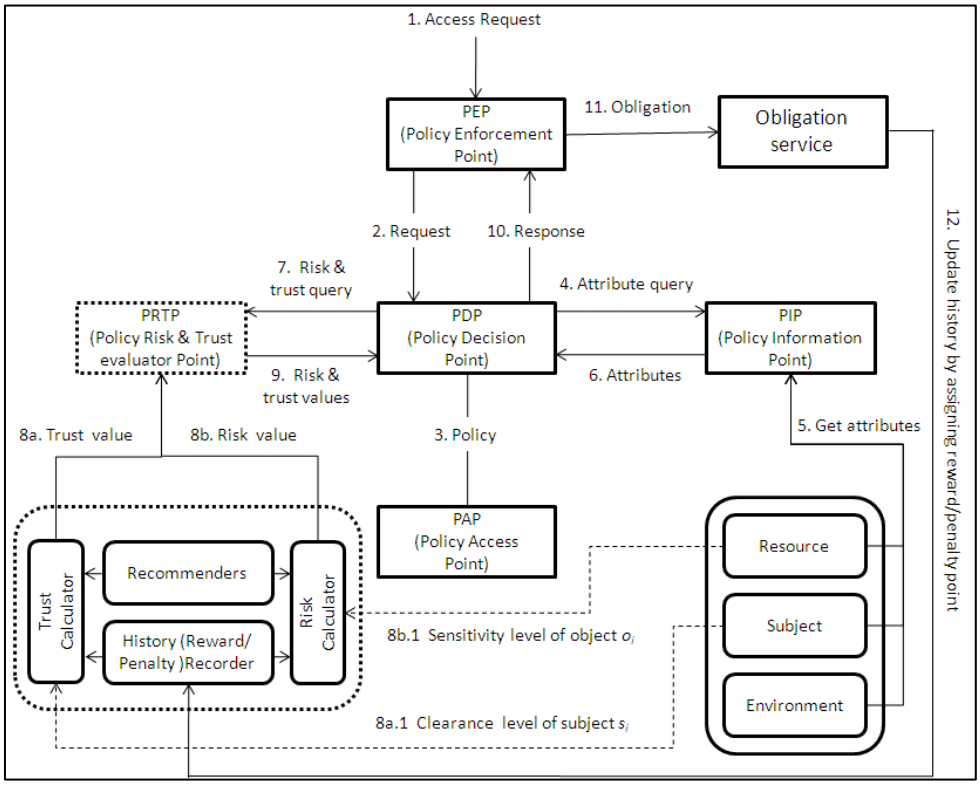


Figure 12 : Flux du processus de la méthode de décision basée sur le risque

La méthode proposée se fait en quatre étapes : La première pour attribuer les points de récompense et de pénalité au sujet-objet, la deuxième pour calculer la confiance, la troisième pour calculer le risque, et la décision est prise à la quatrième étape.

✓ Étape 1 : Attribution des points de récompense et de pénalité

Après que l'accès est donné, une obligation de service est exécutée dans le système qui va décider d'attribuer des points de récompense ou des points de pénalité pour les utilisateurs. En pratique, l'obligation de service (OBL) dépend de l'application. Par conséquent, il est très difficile de concevoir des mécanismes génériques qui pourraient permettre à un système de décider d'attribuer des points de récompense ou de pénalités pour les utilisateurs.

✓ Étape 2: Calcul de la confiance

La méthode de calcul dynamique des valeurs de confiance est conçue pour satisfaire aux exigences suivantes:

- Propriété 1: Si les points de récompense augmentent, alors la valeur de la confiance augmente.
- Propriété 2: Si les points de pénalité augmentent alors la valeur de la confiance diminue.
- Propriété 3: Si ni les pénalités, ni les récompenses sont disponibles, ou seulement les pénalités sont disponibles, alors la valeur de confiance est fixée à une valeur minimale / valeur par défaut, qui est l_s.
- Propriété 4: Si des points de récompense seulement sont disponibles, alors la valeur de confiance est fixée à une valeur maximale, qui est 2 * l_s.

Pour chaque paire d'objet-sujet, on calcule une valeur de confiance (T_v (s, o)).

La confiance est calculée sur la base de deux facteurs:

1) Niveau d'autorisation du sujet (l_s), et

2) L'historique des points de récompense (H^+ (s, o)).

En se basant sur les niveaux d'autorisation, les sujets peuvent être classés de nombreuses façons. Par exemple, dans le cas de la méthode de classification militaire, l'un des niveaux d'autorisation suivants est affecté à un sujet {Top Secret, secret, confidentiel, sensible mais non classifié, non classifié}.

Soit L_S: S → L le niveau d'autorisation maximal que chaque sujet peut avoir. Et soit l_s: s → l le niveau d'autorisation actuel d'un sujet s, qui doit être $l_s \leq L_S$ (L_S doit dominer l_s). Au début, quand il n'y a pas d'historique local adéquat pour le sujet, le système utilisera les conseils des recommandeurs pour calculer l'historique des point de récompense (H^+ (s, o)). Le système peut aussi s'adresser aux recommandeurs lorsqu'il a besoin des informations supplémentaires sur le sujet. Le facteur de l'historique des points de récompense (H^+ (s, o)) représente simplement le pourcentage des points de récompense par rapport au total des points.

Le H^+ (s, o) est calculé de la manière suivante.

$$H^+(s,o) = \begin{cases} w_0\left(\frac{R^I}{R^I + P^I}\right) + \sum_{k=1}^{m} w_k\left(\frac{R_k^E}{R_k^E + P_k^E}\right) & \textit{Si l'histoire est disponible} \\ 0 & \textit{Sinon} \end{cases}$$

Où R^I et P^I représentent, respectivement, le nombre total des points de récompense et des points de pénalité, que le système stocke localement.

R^E et P^E représentent, respectivement, le nombre total des points de récompense et des points de pénalité, qui sont envoyés par les recommandeurs. m représente le nombre total des recommandeurs. Chaque recommandeur peut avoir des valeurs de poids w différents.

Toutefois, la somme de toutes les valeurs des poids $(w_0 + \sum_{k=1}^{m} w_k)$ est 1. Lorsque les recommandations ne sont pas nécessaires, ni disponibles, alors la valeur de w_0 est 1.

Après avoir mesuré le niveau d'autorisation du sujet et l'historique des points de récompense, on calcule la valeur de confiance pour la paire sujet-objet (T_v (s, o)) de la manière suivante :

$$T_v(s,o) = l_s \times [1 + H^+(s,o)]$$

Dans cette équation, le niveau de sensibilité du sujet (l_s) est multiplié avec le facteur 1 + H^+ (s, o). On ajoute 1 à H^+ (s, o), parce que chaque fois que le système n'a pas le record des points de récompense (H^+ (s, o)), alors il définit la valeur de confiance comme étant la valeur par défaut qui est l_s.

Cette équation prouve que la gamme des valeurs de la confiance se situe toujours entre [l_s, 2 × l_s]. En effet, dans le pire des cas, lorsque le sujet s ne dispose pas de points de récompense, alors il reçoit la valeur minimale de confiance. Comme il n'y a pas de points de récompenses, les valeurs de RI et RE deviennent 0. Par conséquent, nous obtenons la valeur de confiance minimale qu'un sujet s peut obtenir qui est la suivante:

T_v (s, o) = l_s × [1 + 0] = l_s.

Dans le meilleur des cas, lorsque le sujet s ne possède pas des points de pénalité, alors le sujet va obtenir une valeur maximale de confiance. Dans ces cas, les valeurs de PI et PE deviennent 0 dans l'équation. Et comme $(w_0 + \sum_{k=1}^{m} w_k) = 1$, on obtient par conséquent, la valeur de confiance maximale qui est : T_v(s, o)= l_s × [1 + 1] = 2 × l_s.

On passe à l'étape de calcul de risque.

✓ Étape 3 : Calcul de risque

La méthode de calcul dynamique des valeurs de risque est conçue pour satisfaire aux exigences suivantes:

- Propriété 5: Si les points de pénalité augmentent, alors la valeur du risque augmente également.
- Propriété 6: Si les points de récompense augmentent, alors la valeur du risque diminue.
- Propriété 7: Si les points de pénalité ne sont pas disponibles, alors la valeur du risque est fixée à une valeur minimale / par défaut, qui est l_o.
- Propriété 8: Si seulement des points de pénalité sont disponibles, alors la valeur du risque est fixée à une valeur maximale, qui est de $2 \times l_o$.

Pour chaque paire de sujet-objet, on calcule une valeur de risque (Rv (s, o)). Le risque est calculé sur la base des deux facteurs suivants:

1) Le niveau de sensibilité de l'objet (l_o), et

2) L'historique des points de pénalité (H^- (s, o)).

En se basant sur la sensibilité, les objets peuvent être classés de plusieurs façons. Par exemple, dans le cas d'une entreprise, un des labels de sensibilité suivants peut être assigné à un objet {Externe, Privée, Sensible, Public}.

L_O: O → L donne le niveau maximal de sensibilité que chaque objet peut avoir.

L_o : o → l donne le niveau actuel de sensibilité d'un objet o, qui doit être $L_o \leq L_O$ (L_O doit dominer L_o).

Le facteur de l'historique des pénalités (H^- (s, o)) représente le pourcentage des points de pénalité par rapport au total des points d'une paire (s, o). Le H^- (s, o) est calculé de la manière suivante.

$$H^{-}(s,o)=\begin{cases} w_0(\frac{P^I}{R^I+P^I})+\sum_{k=1}^{m} w_k(\frac{P_k^E}{R_k^E+P_k^E}) & Si \quad l'histoire \quad est \quad disponible \\ 0 & Sinon \end{cases}$$

Où R^I et P^I représentent, respectivement, le nombre total des points de récompense et des points de pénalité, que le système stocke localement.

R^E et P^E représentent, respectivement, le nombre total des points de récompense et des points de pénalité, qui sont envoyés par les recommandeurs. Le w représente la valeur du poids et la somme de toutes les valeurs de poids $(w_0+\sum_{k=1}^{m} w_k)$ est 1.

Après avoir mesuré le niveau de sensibilité d'un objet et l'historique des point de pénalité, la valeur du risque pour la paire sujet-objet (R_v (s, o)) est calculée de la manière suivante.

$$R_v(s,o)=l_o\times[1+H^{-}(s,o)]$$

Dans cette équation, nous avons multiplié le niveau de sensibilité de l'objet (l_o) avec le facteur 1 + H^-(s, o). Le 1 est ajouté à H^-(s, o), parce qu'à chaque fois que le système ne dispose pas d'un record de points de pénalité (H^-(s, o)), alors la valeur du risque est définie comme étant la valeur par défaut, qui est l_o.

La gamme des valeurs de risque est toujours comprise entre [l_o, 2 × l_o].

- ✓ <u>Étape 4: Le mécanisme de décision</u>

Une fois les valeurs de confiance et de risque sont calculées, le système prendra une décision en se basant sur l'équation ci-dessous.

$$D(T_v(s,o),R_v(s,o))=\begin{cases} Permit \quad Si \quad T_v(s,o)\geq R_v(s,o) \\ Deny \quad Sinon \end{cases}$$

Si la valeur de confiance T_v (s, o) est supérieure ou égale à la valeur de risque R_v (s, o), alors le système va permettre l'accès, sinon la demande d'accès sera refusée.

Dans ce papier, la méthode de décision dynamique basée sur le risque pour les systèmes de contrôle d'accès introduit le concept de l'attribution des points de récompense et de pénalité pour un sujet par rapport à des objets particuliers. Ces récompenses et pénalités reflètent le comportement passé du sujet. Sur la base des comportements passés et des niveaux actuels de sécurité d'un sujet et d'un objet, les valeurs de confiance et de risque sont calculées et associées à chaque paire objet-sujet. En se basant sur ces valeurs, une décision d'accès est prise.

Ce travail peut être amélioré en l'étendant afin d'intégrer d'autres paramètres dans les méthodes de calcul de confiance et du risque comme les propriétés des sujets. Comme on peut travailler sur les niveaux de sensibilité des objets et des autorisations des sujets, on peut proposer des méthodes afin de rendre ces valeurs calculables.

2.4.11 Contrôle d'accès à base de rôle avec risque RBAC^R

Dans l'article [17], un nouveau modèle de contrôle d'accès RBAC^R (Role Based Access Control Model with risk) a été introduit, dans lequel chaque décision de contrôle d'accès est prise après l'évaluation du risque.

La méthode de l'évaluation des risques proposée considère un ordre partiel sur les objets et sur les actions à saisir pour capturer la notion de l'importance des objets et la criticité des actions, et détermine le risque d'attribuer un rôle spécifique à un utilisateur spécifique. Le cas de la délégation de rôles est également considéré.

Dans le modèle RBAC^R, l'évaluation des risques est examinée et appliquée pour évaluer les valeurs des risques associés aux demandes d'accès dans un système RBAC donné. Les fonctions d'analyse des risques devraient jouer un rôle important dans le raisonnement et la prise de décision dans de tels systèmes. En ajoutant le paramètre d'évaluation des risques aux systèmes RBAC, nous obtenons le modèle RBAC^R.

Dans le modèle RBAC^R, la prise de décision dépend de l'évaluation des risques. Les fonctions de risque définies, RF, est un élément majeur dans ce modèle et le rend différent du modèle RBAC classique.

Pour définir l'ensemble des fonctions d'analyse des risques, il faut identifier les situations qui peuvent conduire à des risques, et ensuite trouver des fonctions appropriées pour calculer les valeurs du risque liées à une situation reconnue. Dans la suite, nous discutons la façon proposée

dans l'article pour définir les fonctions appropriées d'analyse des risques pour les deux ingrédients principaux dans RBAC : l'attribution des rôles et la délégation.

L'idée de base en ce qui concerne l'évaluation des risques pour l'attribution des rôles est la suivante:

- Pour chaque utilisateur u, on assigne un niveau de confiance, noté CNF (u).
- Pour chaque rôle R, nous calculons le niveau minimal de confiance nécessaire pour l'attribution du rôle, noté MLC (R).

Ensuite, la valeur de risque de l'attribution du rôle $0 \leq RV(u, R) \leq 1$, est calculée par la formule suivante:

$$RV(u,R) = \begin{cases} 0, & Si \quad CNF(u) \geq MLC(R) \\ 1 - \dfrac{CNF(u)}{MLC(R)}, & Sinon \end{cases}$$

Dans la formule de calcul du risque ci-dessus, on suppose que les niveaux de confiance pour les utilisateurs et les niveaux minimaux de confiance nécessaires pour des rôles devraient se situer dans le même domaine, tel que le domaine [0..3] (par exemple, le domaine utilisé pour exprimer les niveaux de sécurité dans un système est, non classifié = 0, restreint = 1, secret = 2, top secret = 3). Par conséquent, nous avons toujours $0 \leq CNF(u) \leq 3$ et $0 \leq MLC(R) \leq 3$. En général, on a le domaine $D = \{0,\ldots, k\}$, de telle sorte que, pour tout $u \in U$ et tout $R \in R$, nous avons $0 \leq CNF(u) \leq k$ et $0 \leq MLC(R) \leq k$. Avec U l'ensemble des utilisateurs, et *R* l'ensemble des rôles.

Ainsi, selon la formule ci-dessus, afin d'obtenir la valeur du risque RV(u, R), on doit calculer MLC (R) pour chaque rôle R.

Une permission est définie comme une paire composée d'une action et d'un objet, et un rôle est un ensemble de permissions. On considère que l'ensemble des actions et l'ensemble des objets sont partiellement ordonnés.

Cet ordre capture la notion de criticité pour les actions et d'importance pour les objets. On peut ensuite calculer l'ordre partiel sur les rôles. Puis, nous pouvons obtenir le niveau minimal de confiance (MLC) requis pour le rôle.

Soient (A = {ai | i = 0, 1…, n}, $\subseteq_a$) et (O ={oi | i = 0, 1…, m}, $\subseteq_o$), respectivement, des ensembles partiellement ordonnés des actions et des objets, avec A l'ensemble des actions et O l'ensemble des objets dans le système. La relation a′ $\subseteq_a$ a, signifie que l'action a' est moins critique que l'action a. De même, o′$\subseteq_o$ o signifie que l'objet o' est moins important que l'objet o.

Par exemple, disons a1 = modify, a2 = write, a3 = move, et a4 = read. On peut considérer un ensemble d'actions classées en fonction de la criticité. Par exemple l'action «modify» est considérée plus critique que les actions «write» et et l'action «move», et l'action «read» est moins critique que l'action «write», ainsi que l'action «move». Cependant, «write» et «move » n'ont aucun rapport entre eux. Ces faits sont décrits par les relations suivantes a2 $\subseteq_a$ a1, a3 $\subseteq_a$ a1, a4 $\subseteq_a$ a2 et a4 $\subseteq_a$ a3, mais a2 et a3, ne sont pas comparable.

La figure 13 présente un ordre partiel pour un système contenant l'ensemble des actions {a1, a2, a3, a4} et l'ensemble des objets {o1, o2, o3, o4}.

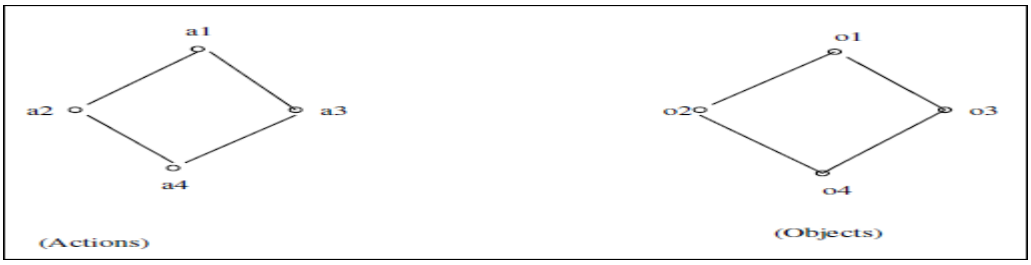


Figure 13 : Actions et objets

A × O est l'ensemble de toutes les permissions. À partir des ensembles ordonnés (A, $\subseteq_a$) et (O, $\subseteq_o$), on déduit une relation d'ordre $\subseteq_{ao}$ définit comme suit :

(a′, o′) $\subseteq_{ao}$ (a, o) ↔ a′ $\subseteq_a$ a ∧ o′$\subseteq_o$ o où (a′, o′) $\subseteq_{ao}$ (a, o) signifie que la permission (a', o') est moins critique que la permission (a, o).

Et comme un rôle R est un sous-ensemble de A × O, alors pour tout rôle R, (R, $\subseteq_{ao}$) est également un poset (un ensemble muni d'une relation d'ordre). Dans la figure 14, nous montrons toutes les autorisations et deux rôles R1 et R2 du système de la figure 13.

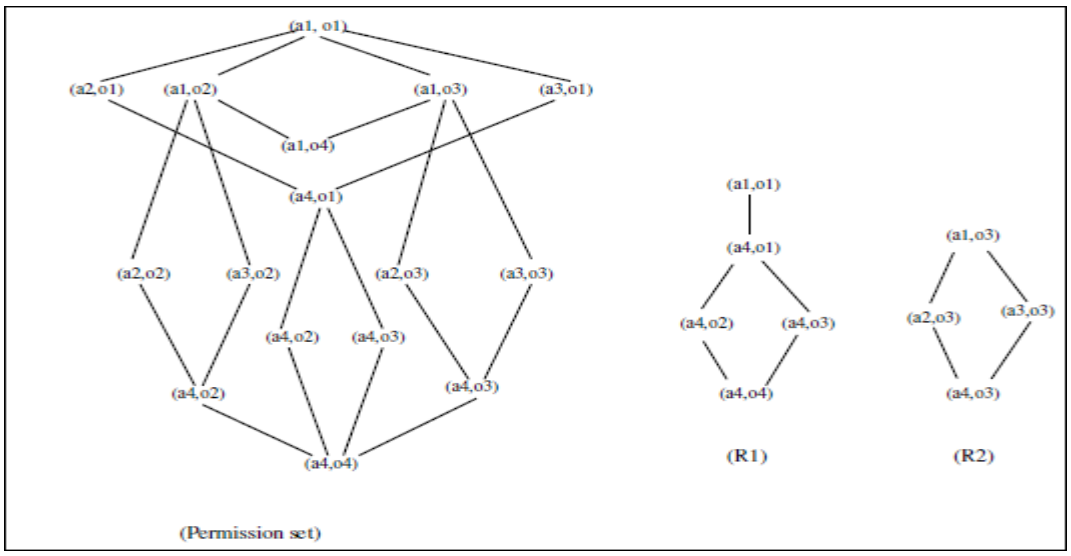


Figure 14 : L'ensemble des permissions et exemples de rôles

Afin de calculer le MLC d'un rôle, nous introduisons d'abord les définitions suivantes:

- Une chaîne dans un rôle R est un sous-ensemble C de R ayant un ordre total, c'est à dire pour tous les (a, o), (a', o') ∈ C, (a, o) $\subseteq_{ao}$ (a', o') ou (a', o') $\subseteq_{ao}$ (a, o). Ainsi, tous les nœuds dans une chaîne C sont comparables. Une chaîne dans un rôle peut être représentée par un chemin avec des arêtes entre les nœuds adjacents.

- La longueur d'une chaîne est le nombre d'arêtes reliant deux nœuds dans la chaîne. Soit l(C) la longueur de la chaîne C, et n(C) le nombre des nœuds de la chaîne C. l(C) = n(C) - 1. Par exemple, la figure 14, (R1), (a4, O4) est la plus courte chaîne qui contient un seul nœud et sa longueur est égale à 0; (a4, o4) - (a4, o2) est une chaîne qui contient deux nœuds, sa longueur est de 1, et (a4, o4) - (a4, o2) - (a4, o1) est une chaîne qui contient trois nœuds, et sa longueur est de 2. Soit C_R l'ensemble de toutes les chaînes dans un rôle $(R, \subseteq_{ao})$. MLC (R) est définie comme étant la longueur de la chaîne la plus longue dans R, à savoir: $MLC(R)=\max\{l(C)|C \in C_R\}$.

Par exemple, pour les rôles décrits dans la figure 14, nous avons MLC (R1) = 3 et le MLC (R2) = 2.

Le risque de délégation est aussi un point important dans l'analyse des risques. Soit del_rv (u1, u2) la valeur du risque que l'utilisateur u1 délègue une de ses permissions à

l'utilisateur u2. Le risque d'une délégation dépend du niveau de confiance du délégant et du délégataire. Plus formellement, on peut avoir:

$$del\_rv(u1,u2) = \begin{cases} 0, & Si \quad CNF(u2) \geq CNF(u1) \\ 1 - \dfrac{CNF(u2)}{CNF(u1)}, & Sinon \end{cases}$$

Pour définir les règles de permission et de délégation on définit d'abord les prédicats suivants:

- permit (r, a, o, c): Le rôle r a la permission de mener une action a sur l'objet o dans le contexte c.
- holds (u, r): utilisateur u a le rôle r.
- delegate (u1, u2, a, o, c): L'utilisateur u1 délègue à l'utilisateur u2 sa permission de mener une action a sur l'objet o dans le contexte c.
- user-permit (u, a, o): L'utilisateur u est autorisé à mener une action a sur l'objet o.

On définit aussi une relation entre les actions et une relation entre les objets, les deux notée <=. a <= a' signifie que a est une sous-action de a', par exemple modifier <= écriture. o <= o' signifie que l'objet o est inclus dans l'objet o'.

La règle de la permission avec RBAC est formalisée comme suit :

$$\frac{permit(r,a',o',c) \in P(\varepsilon), \quad \varepsilon \subset c, holds(u,r,) \quad a \leq a', \quad o \leq o'}{user\_permit(u,a,o)}$$

Où ε est l'environnement, et P(ε) est l'ensemble de permissions dans l'environnement.

La règle de la permission stipule que si la permission permit (r, a', o', c) est dans P (ε) et l'utilisateur u tient le rôle r alors pour toute sous-action a de a'(a<= a') et o sous-objet de o '(o <= o'), l'utilisateur u est autorisé à mener l'action a sur l'objet o.

Ajoutons la notion de l'évaluation de risque à la règle de la permission de RBAC.

La règle de la permission avec l'évaluation des risques: est formalisé comme suit:

$$\frac{permit(r,a',o',c) \in P(\varepsilon), \quad \varepsilon \subset c, \quad holds(u,r), \quad rv(u,r) \leq risk\_thresholds(\varepsilon,a,o), \quad a \leq a', \quad o \leq o'}{permit\_with\_risk(u,a,o,rv(u,r))}$$

Où le prédicat permit_with_risk (u, a, o, rv) exprime le fait que l'utilisateur u est autorisé à mener une action a sur un objet o avec une valeur de risque rv. Et risk threshold(ε, a, o) est utilisé pour spécifier une valeur de seuil pour le risque.

La règle de la permission avec risque stipule que l'utilisateur u n'est autorisé à mener une action a sur un objet o avec une valeur de risque de rv(u, r) que si cette valeur est inférieure à un seuil donné par la fonction risk_threshold(ε, a, o).

La règle de la délégation avec RBAC est formalisée comme suit :

$$\frac{delegate(u1,u2,a',o',c) \in D(\varepsilon), \quad \varepsilon \subset c, \quad user-permit(u1,a',o'), \quad a \leq a', \quad o \leq o'}{user\_permit(u2,a,o)}$$

Où D (ε) est l'ensemble des délégations dans l'environnement ε et u1, u2 ∈ U, u1 est le délégant, u2 est le délégataire, et c est le contexte.

La règle de la délégation stipule que: Si une délégation delegate(u1, u2, a', o', c) est dans D (ε), et dans l'environnement ε, et l'utilisateur u1 dispose de la permission (a',o'), alors pour tout sous-action a de a', et tout sous-objet o de o', l'utilisateur u1 peut déléguer à l'utilisateur u2 sa permission de mener une action a sur un objet o.

La règle de la délégation avec l'évaluation des risques: est formalisé comme suit:

$$\frac{\begin{array}{c} delegate(u1,u2,a',o',c) \in D(\varepsilon), \quad \varepsilon \subset c, \quad permit\_with\_risk(u1,a',o',t) \\ t + del\_risk(u1,u2) \leq risk\_threshold(\varepsilon,a,o), \quad a \leq a', \quad o \leq o' \end{array}}{permit\_with\_risk(u2,a,o,,t + del\_risk(u1,u2))}$$

En effet, supposons qu'un utilisateur u1 peut effectuer une action a sur un objet o avec une valeur de risque t et que l'utilisateur u1 peut déléguer cette autorisation à un autre utilisateur u2, alors la valeur de risque que l'utilisateur u2 exécute cette action a sur l'objet o peut être supérieur à t, c'est-à-dire (t + del_risk (u1, u2))> t.

Maintenant, avec ces règles, nous pouvons raisonner sur l'affectation des autorisations avec l'évaluation des risques.

Le problème avec ce modèle, c'est au niveau du calcul du risque de l'affectation d'un rôle à un utilisateur. Ce calcul se fait en fonction de CNF(u) (le niveau de confiance à un utilisateur), cette valeur ne doit pas être une simple entrée mais elle doit être calculable en fonction des paramètres du contexte.

La notion de confiance est largement utilisée dans la sécurité des systèmes informatiques. Par exemple, les entités participant à une transaction de commerce électronique doivent avoir «confiance» les uns aux autres ou compter sur une tierce partie à laquelle ils ont confiance. Cependant, il n'y a pas de formalismes acceptées ou techniques pour la spécification de la confiance. Les systèmes de sécurité ont été construits sous l'hypothèse que les concepts comme «fiabilité» ou «confiance» sont bien compris, malheureusement, sans même s'entendre sur ce que la «confiance» signifie, comment la mesurer, comment comparer deux valeurs de confiance. Cela crée un certain nombre d'ambiguïtés dans la construction des systèmes sécurisés.

Considérons, par exemple, une base d'informations opérationnelles dans une grande société. Elle est générée par l'accumulation des informations provenant de plusieurs sources. Certaines de ces sources sont sous le contrôle direct de l'administration de la société et sont donc considérées comme dignes de confiance. D'autres sources sont des sources «amicales» et les informations provenant directement d'elles sont également considérées comme dignes de confiance. Cependant, ces sources «amicales» peuvent avoir des informations qui proviennent des sources dont la société n'a la moindre connaissance, donc la société n'a aucun fondement réel pour déterminer la qualité. Ce sera plutôt 'naïf' de la part de la société de faire confiance à ces informations de la même façon qu'elle le fait aux informations fournies à partir des sources sous son contrôle direct. En même temps, ne pas faire confiance à toutes ces informations les rend inutiles. Les modèles binaires de confiance existants (où la confiance n'a que deux valeurs, «aucune confiance» et «confiance totale» et qui sont les plus largement utilisés dans les systèmes informatiques) vont, néanmoins, classer la valeur de ces informations à l'un de ces deux niveaux.

Ceci nous a motivé pour trouver et proposer un nouveau modèle de confiance pour pouvoir calculer le CNF(u) du modèle $RBAC^R$.

2.4.12 *Modèle de calcul de confiance entre deux entités (le « Truster » et le « Trustee »)*

Pour surmonter la limitation du $RBAC^R$, on introduit dans ce qui suit l'article [18] qui traite le problème de calcul de confiance entre deux entités.

Les auteurs de l'article proposent un modèle qui nous permet de formaliser les relations de confiance. La relation de confiance entre un «truster» (L'entité qui fait confiance à l'entité cible) et un «trustee» (L'entité cible qui est digne de confiance) est associée à un contexte et dépend de l'expérience, des connaissances, et de la recommandation que le «truster» a à l'égard de «trustee» dans un contexte donné.

Nous montrons comment ce modèle permet de mesurer la confiance et comparer deux relations de confiance dans un contexte donné.

Dans ce modèle la relation de confiance entre le «truster» et le «trustee» peut être de différents degrés. Par ailleurs, dans le modèle, une relation de confiance entre un «truster» et un « trustee » n'est jamais absolue. Le «truster» fait confiance au «trustee» à l'égard d'un certain contexte. La confiance entre un «truster» et un «trustee» dans un contexte spécifique est définie dans le modèle comme un vecteur à 3 éléments. Ces éléments correspondent à des facteurs, à savoir, l'expérience, les connaissances et la recommandation, desquels dépend la relation de confiance. La valeur de chacun de ces éléments est représentée comme un triplet (b, d, u) où b, d et u désignent, respectivement, la croyance (belief), l'incrédulité (disbelief), et l'incertitude (uncertainty).

Ce modèle n'est pas très utile dans le calcul de la confiance lorsque le «truster» n'a pas d'expérience, des connaissances, ou des recommandations au sujet d'un «trustee» dans un contexte donné. Un exemple permettra d'illustrer ce point.

Supposons qu'un utilisateur A (truster) n'a pas d'expérience, des connaissances ou de recommandation en ce qui concerne le développeur du logiciel B (trustee) à l'égard de l'élaboration des logiciels antivirus (le contexte). Le modèle ne sera pas en mesure d'évaluer la relation de confiance dans ce cas. En supposant que l'expertise pour développer un logiciel anti-spam (un contexte lié) est similaire à l'expertise nécessaires pour développer des logiciels anti-virus, il semble naturel que le «truster» A sera en mesure de déterminer combien faire confiance à B pour le (différent mais connexe) contexte. Le modèle peut être étendu pour s'adapter à cette situation.

Les auteurs adoptent la définition de la confiance fournie par Grandison et Sloman dans leur ouvrage intitulé «A survey of trust in internet applications».

La confiance est définie comme la croyance ferme à la compétence d'une entité d'agir de manière fiable, sûre et sécurisée dans un contexte spécifié.

Dans le même ouvrage, Grandison et Sloman définissent la méfiance comme le «manque de conviction à la compétence d'une entité d'agir de façon fiable, sûre et sécurisée». Cependant, les auteurs croient que la méfiance est un peu plus forte que juste un «manque de croyance», et choisissent d'être plus précis et de définir la méfiance comme suit :

La méfiance est définie comme l'incrédulité ferme à la compétence d'une entité d'agir de façon fiable, sûre et sécurisée dans un contexte spécifié.

Dans le modèle, la confiance est spécifiée comme une relation de confiance entre un «truster», disons entité A, une entité qui fait confiance à l'entité cible, et un «trustee», disons entité B, l'entité qui est digne de confiance. Le truster A est toujours une entité active (par exemple, un être humain ou un sujet). Le trustee B peut être soit une entité active ou une entité passive (par exemple, un élément d'information ou un logiciel).

La relation de confiance entre un truster, A, et un trustee, B, n'est jamais absolue [Grandison et Sloman 2000]. Le «truster» fait, toujours, confiance au «trustee» à l'égard de sa capacité à exécuter une action spécifique ou de fournir un service spécifique. Par exemple, une entité A peut faire confiance à une autre entité B par rapport à la capacité de ce dernier de garder un secret. Toutefois, cela ne signifie pas que, si A veut réaliser un travail efficace, A va faire confiance à B pour le faire. De même, si nous voulons comparer deux valeurs de confiance, nous ne pouvons pas comparer deux valeurs de confiance arbitrairement. Nous avons besoin de comparer les valeurs de confiance qui servent des fins similaires. Cela nous amène à associer une notion de contexte à la relation de confiance. Des exemples de contextes sont (i) de fournir un service, (ii) de prendre des décisions au nom d'un truster, et (iii) d'accéder aux ressources d'un truster.

Dans ce modèle de confiance les auteurs ont adapté le modèle d'opinion de [19]. Dans ce travail, l'opinion est représentée comme un triplet (b, d, u) où b représente la croyance, d représente l'incrédulité et u représente l'incertitude. Chacune de ces composantes a une valeur comprise entre [0,1] et la somme des trois composantes est 1.

Ce modèle sera adopté afin de représenter la confiance du truster A au trustee B dans un certain contexte c comme un triplet ($_Ab^c_B$, $_Ad^c_B$, $_Au^c_B$), où $_Ab^c_B$ est la croyance de A en B dans le contexte c, $_Ad^c_B$ est l'incrédulité de A en B dans le contexte c, et $_Au^c_B$ est l'incertitude du truster A en B dans le contexte c.

La relation de confiance n'est pas statique mais évolue avec le temps. Même s'il n'y a pas de changement dans les facteurs qui influent la confiance, dans une période de temps, les valeurs de la relation de confiance à la fin de la période ne sont pas les même que celles au début de la période. Ainsi, nous avons besoin de spécifier le temps pour décrire une relation de confiance. La relation de confiance entre A et B dans un contexte c au temps t est formellement notée $(A \xrightarrow{c} B)_t$.

La relation de confiance $(A \xrightarrow{c} B)_t$ est une matrice 3 * 3. Les lignes de la matrice correspondent aux trois paramètres, à savoir, l'expérience, les connaissances, et la recommandation, desquels dépend la confiance. Chacun de ces paramètres est représenté en terme de (b, d, u) où b signifie la croyance au paramètre pour l'évaluation de la confiance, d spécifie l'incrédulité au paramètre, et u signifie l'incertitude au paramètre pour évaluer la confiance. Ces trois termes constituent les colonnes de la matrice de la confiance.

Les trois paramètres peuvent ne pas avoir la même importance pour l'évaluation de la confiance. Le vecteur de la politique de confiance spécifie le facteur de normalisation qui donne le poids relatif de chaque paramètre. En appliquant le facteur de normalisation à la relation de confiance cela donne une relation de confiance normalisée.

La relation de confiance normalisée entre le truster A et le trustee B concernant un contexte c au temps t est formellement notée $(A \xrightarrow{c} B)_t^N$. Cette confiance normalisée est représentée comme un seul triplet $(_A\hat{b}^c{}_B, {}_A\hat{d}^c{}_B, {}_A\hat{u}^c{}_B)$.

Définissions formellement les trois paramètres de la relation de confiance, à savoir, l'expérience, les connaissances, et la recommandation.

2.4.12.1 L'expérience

L'expérience d'un « truster » à propos d'un «trustee» est définie comme la mesure de l'effet cumulatif d'un certain nombre d'événements qui ont été rencontrées par le «truster» à l'égard du «trustee» dans un contexte particulier et pendant une période de temps spécifiée.

Pour un périphérique, cela pourrait être des incidents comme le nombre de défauts rencontrés, les occurrences des trafics, la qualité des données recueillies, la réactivité aux alarmes et aux signaux de contrôle. Pour un utilisateur humain, cela pourrait être les décisions prises dans le passé, le temps pris pour l'exécution des tâches, etc.

La confiance d'un «truster» en un «trustee» peut changer en raison des expériences du « truster » avec le «trustee» dans un contexte particulier. L'expérience dépend du type des événements, à savoir, positifs, négatifs ou neutres, qui ont été rencontrés par le «truster».

2.4.12.2 Les connaissances

La confiance d'un «truster» à un «trustee» peut changer en raison de certaines connaissances que le «truster» vient de posséder en ce qui concerne le «trustee» dans un contexte particulier. Les connaissances peuvent être de deux types : connaissances directes et indirectes.

Les Connaissances directes sont celles que le «truster» acquiert par lui-même. Elles peuvent être obtenues par le « truster » dans quelque temps auparavant pour une certaine raison, elles peuvent être des éléments d'informations sur le «trustee» pour lequel le «truster» a une preuve concrète pour être vrai.

Les connaissances indirectes, d'autre part, sont les informations que le «truster» n'acquiert pas par lui-même. La source de la connaissance indirecte est la réputation du «trustee» dans le contexte. Le «truster» peut avoir une idée de la réputation du «trustee» de diverses sources comme les revues, les journaux, les bulletins de nouvelles, les opinions des gens etc.

2.4.12.3 La recommandation

Une recommandation pour un «trustee» est définie comme une mesure d'un jugement subjectif ou objectif d'un recommandeur du «trustee» au «truster».

Les recommandations en cas d'un périphérique peuvent être fournies par d'autres organisations qui ont utilisé ce périphérique dans des circonstances similaires. Pour un utilisateur

humain, les recommandations, par exemple, peuvent être fournies par des organisations dans lesquelles cet utilisateur a travaillé dans le même (ou similaire) contexte.

La valeur de la confiance d'un «truster» en un «trustee» peut changer en raison d'une recommandation pour le «trustee». Un recommandeur envoie son avis, en termes d'un triplet (b,d,u), dans un contexte spécifié . Par ailleurs, la recommandation peut être obtenue par le « truster » de plus d'une source.

2.4.13 *Modèle de contrôle d'accès basé sur la confiance pour les applications d'informatique ubiquitaire*

Dans leur article [20], les auteurs ont adapté le modèle qu'on vient de présenter (Context-sensitive trust model) afin de proposer un modèle RBAC basé sur la confiance pour les systèmes d'informatique ubiquitaire. On représentera dans ce qui suit ce travail.

Les utilisateurs (humains ou périphériques) sont évalués pour leur fiabilité avant qu'ils ne soient affectés à des différents rôles. Les rôles sont associés à une gamme de confiance indiquant le niveau de confiance minimal qu'un utilisateur a besoin d'atteindre avant qu'il puisse être affecté à ce rôle. Une autorisation est également associée à une gamme de confiance indiquant le niveau de confiance minimal qu'un utilisateur, affecté à un certain rôle, a besoin d'atteindre pour activer l'autorisation.

Initialement, une entité A ne fait pas une confiance totale à une nouvelle entité B. L'entité A a besoin d'évaluer une relation de confiance avec l'entité B dans un certain contexte. Le contexte de ce modèle est le rôle d'un utilisateur qui lui sera assigné.

Nous ferons référence au contexte avec contexte-rôle rc. Les utilisateurs peuvent être associés à des rôles multiples. Afin de déterminer l'autorisation entre un utilisateur et un rôle, la valeur de la confiance d'un utilisateur est évaluée en fonction de chaque contexte-rôle séparément. La relation de confiance entre l'utilisateur humain ou périphérique, et le système dans le contexte-role rc dépend de trois facteurs: les propriétés, l'expérience et les recommandations.

La relation de confiance entre le «truster», A, et le «trustee», B, dans un certain contexte-rôle rc est représentée formellement comme un triplet $({}_{A}b^{rc}{}_{B}, {}_{A}d^{rc}{}_{B}, {}_{A}u^{rc}{}_{B})$, où ${}_{A}b^{rc}{}_{B}$ est la croyance de A en B au sujet de la fiabilité de ce dernier, ${}_{A}d^{rc}{}_{B}$ est l'incrédulité de A en B et ${}_{A}u^{rc}{}_{B}$ est

l'incertitude de A en B. Chacune de ces composantes a une valeur comprise entre [0,1] et la somme de ces éléments est 1.

Ce modèle de confiance nous permettra de résoudre les limitations du modèle de contrôle d'accès $RBAC^R$ en calculant le niveau de confiance des utilisateurs puisque cette valeur de confiance peut jouer le rôle de la valeur de CNF(u) du modèle de contrôle d'accès $RBAC^R$.

2.5 Conclusion

Les modèles traditionnels de contrôle d'accès (RBAC, MAC, DAC …) sont rigides. Les décisions de contrôle d'accès ne peuvent être que d'autoriser ou refuser l'accès. Avec ces modèles on considère que les autorisations d'accès sont connues à l'avance et que les règles ont été mises en place correctement, mais dans des contextes réels, les erreurs sont commises et des situations imprévues ou d'urgence peuvent se produire.

Généralement lorsqu'une opération doit être effectuée, un gestionnaire de contrôle d'accès doit analyser la requête et décider s'il s'agit d'une opération légale ou non. Pour cela, la décision peut être prise de deux façons :

- Prise explicitement par un administrateur dans le cadre d'une politique d'accès, la requête est autorisée auquel cas l'opération peut se dérouler sans problèmes ou refusée dans le cas contraire.
- Prise suite à une décision qui doit être calculée suivant diverses variables. Dans ce cas, nous parlons de gestion de risque.

Afin de rendre RBAC dynamique et flexible et en se basant sur son extensibilité, plusieurs propositions de modèles ont vu le jour et qui ont proposé l'intégration de fonctionnalités supplémentaires.

On a vu des modèles qui ont intégré des paramètres de contexte comme GRBAC [7] et DRBAC [8] sauf que le processus décisionnel n'est pas aussi puissant que celui des modèles qui ont intégré le processus de gestion de risque.

On a vu aussi les modèles TCAC [9] et T_SR [10] qui ont intégré la notion de la confiance et du contexte: L'accès n'est autorisé que lorsque l'utilisateur atteint un certain niveau de confiance et que les contraintes du contexte sont satisfaites. Ces modèles peuvent être enrichis en calculant

dynamiquement le niveau de confiance que l'utilisateur doit atteindre. Aussi on peut ajouter la notion de risque pour permettre l'accès dans des situations imprévue ou d'urgence.

On a vu quelques travaux qui ont intégré la notion de risque : CRAAC [21] calcule le risque en comparant le niveau de confiance du demandeur d'accès avec le niveau de sensibilité de l'objet à accéder. CRAAC s'est limité aux propriétés des utilisateurs pour calculer leurs niveaux de confiance. Alors que le niveau de confiance peut être calculé à partir de plusieurs autres variables.

Aussi on a présenté le travail [16] qui s'est éloigné de RBAC et de la notion des rôles et qui propose une méthode pour calculer dynamiquement les valeurs de la confiance et du risque pour chaque paire sujet-objet. Le calcul des valeurs de confiance et de risque se basent essentiellement sur l'historique des utilisateurs sauf qu'on peut utiliser d'autres paramètres. Aussi on trouve qu'il est préférable de travailler sur RBAC. En effet, RBAC, en utilisant le rôle comme intermédiaire entre les sujets et les permissions, facilite et simplifie les tâches d'administration en diminuant le nombre d'affectations à manipuler. En plus RBAC est suffisamment mature pour une utilisation globale. L'utilisation massive de ce système depuis de nombreuses années montre également qu'il est robuste et prêt à être utilisé dans toutes les branches de l'industrie.

On a vu aussi le modèle $RBAC^R$ [17], où l'accès est autorisé sans risque si le niveau de confiance de l'utilisateur est supérieur ou égal au niveau de confiance exigé par le rôle et avec risque dans le cas contraire mais il faut que la valeur de risque ne dépasse pas un certain seuil. Ce modèle peut être amélioré en proposant une méthode qui permet de calculer dynamiquement les niveaux de confiance des utilisateurs et les niveaux de confiance des rôles et de prendre en considération le contexte pour déterminer le seuil acceptable de risque.

Nous aussi nous avons profité de l'extensibilité de RBAC pour proposer un nouveau modèle qui sera présenté dans le chapitre suivant. Dans ce modèle on a retenu les points forts des modèles qu'on a étudiés dans ce chapitre.

En effet, dans notre modèle le contrôle d'accès est basé sur le risque. L'estimation de risque est calculée dynamiquement en se basant sur les niveaux de confiances des utilisateurs, les niveaux de confiance exigés par les rôles et le contexte. Pour la méthode de calcul de risque on s'est inspiré de la formule du modèle $RBAC^R$ [17].

Les niveaux de confiance sont aussi calculés dynamiquement en fonction de certains paramètres essentiellement l'historique des utilisateurs. Pour ce faire nous avons adopté le modèle de calcul de confiance entre deux entités le «Truster » et le « Trustee » proposé dans l'article [18] et qui a été adapté au modèle RBAC dans l'article [20].

Enfin, afin de rendre notre modèle plus flexible et lui permettre de bien fonctionner dans les environnements dynamiques, la décision de contrôle d'accès est prise en 3 étapes : étape de l'affectation aux rôles, étape de l'activation des rôles et étape de l'exécution des permissions. A chaque étape un modèle de gestion de risque : Un utilisateur affecté à un certain rôle peut être interdit de l'activer dans certains contextes, et s'il arrive à l'activer on peut lui refuser d'exécuter certaines permissions. On parle ici de RBAC avec conditions.

Chapitre 3 : Le modèle de contrôle d'accès basé sur les rôles et le risque

Après avoir présenté notre recherche bibliographique et introduit les différentes modélisations de contrôle d'accès, on s'intéressera dans ce chapitre à présenter notre contribution qui consiste en l'élaboration d'un modèle solide de contrôle d'accès permettant d'éviter les failles de sécurité qu'on peut avoir avec d'autres modèles traditionnels. Ce nouveau modèle est basé sur RBAC et il propose une nouvelle technique d'évaluation de risque pour pouvoir faire face aux besoins vastes et complexes des nouveaux systèmes informatiques.

3.1 Introduction

Le concept de rôle est central dans un système RBAC. En effet, un utilisateur n'est autorisé à mener une action sur une ressource que si son rôle lui permet la faire. A chaque rôle est associé un certain nombre de privilèges qui peuvent varier avec le temps dépendamment de la politique de sécurité mise en place. Par ailleurs, à chaque instant un utilisateur peut être affecté à un ou plusieurs rôles. Il est utile de noter qu'un système RBAC est hautement dynamique du fait de la modification continuelle des privilèges associés aux rôles et le changement fréquent d'affectation des utilisateurs aux rôles.

En effet, une grande partie des failles des systèmes sont dues à cette modification continue : proposer une méthode d'estimation des risques permet de réduire les erreurs, améliorant ainsi la sécurité des systèmes.

Nous étudions un moyen de gérer les risques pour le modèle RBAC, dans les environnements dynamiques, basé sur un modèle de confiance tout en tenant compte du seuil du risque de l'environnement qui ne doit pas être dépassé.

De toute évidence, l'étape critique des modèles inspirés de RBAC et qui se basent sur le risque est l'estimation du risque d'une demande d'accès qui consiste à la possibilité d'une fuite

d'informations dans l'avenir. Quelle que soit la demande d'accès qui doit être autorisée cela dépend uniquement de l'estimation du risque. Le but est de déterminer la possibilité des divulgations future de l'information résultantes de l'accès actuel.

3.2 Motivation

Les mécanismes traditionnels de contrôle d'accès sont insensibles au contexte, donc ils ne peuvent pas garantir une bonne sécurité dans un environnement distribué et dynamique comme l'environnement d'informatique ubiquitaire.

Les recherches actuelles sur le contrôle d'accès, déjà introduites dans le chapitre précédent, sont principalement basées sur le contexte et le rôle. Certaines utilisent la confiance comme la composante fondamentale. Certaines combinent la confiance avec le risque pour créer un service de sécurité renforcé.

Aujourd'hui, le modèle de contrôle d'accès basé sur les rôles (RBAC) est largement discuté et appliqué dans la sécurité informatique. Dans RBAC, la décision d'accès dépend des rôles des utilisateurs. Par exemple, un utilisateur avec le rôle «comptable» a, normalement, des droits d'accès différents de ceux d'un utilisateur avec le rôle «gestionnaire». Le processus d'attribution des rôles est généralement basé sur une analyse approfondie des risques. En outre, comme la perception du risque change dans le temps, les politiques de contrôle d'accès peuvent également être modifiées dynamiquement. Par conséquent, étudier les méthodes de gestion des risques et les techniques appliquées pour RBAC dans des environnements dynamiques est un domaine de recherche actif.

Le risque est le dommage potentiel découlant de certains processus actuels ou de certains événements futurs. Il est souvent associé à la probabilité de certains événements qui sont considérés comme indésirables.

L'évaluation des risques est un outil efficace dans la prise de décision.

Le «Risque» est un terme souvent associé à des connotations négatives. Il est défini, par la norme australienne et néo-zélandaise de gestion des risques - AS / NZS 4360:2004 (Joint Technical Committee OB-007 2004), comme étant «la chance que quelque chose se produise et qui aura un impact sur les objectifs ». Comme la norme fait également remarquer, « le risque est mesuré en termes d'une combinaison des conséquences d'un événement et de leurs probabilité».

Cette définition assez générale des risques prend en considération le fait que les activités impliquant des risques peuvent souvent avoir des résultats positifs ainsi que négatifs. En tant que tel, la gestion de risque n'est pas seulement une question de réduction des conséquences négatives, mais aussi augmentation des résultats positifs.

La gestion des risques est une procédure clé qui est largement utilisée par les organisations publiques et privées du monde entier pour améliorer l'efficacité opérationnelle. Le processus de gestion des risques a été normalisé dans AS/NZS 4360:2004 et il s'agit essentiellement de cinq activités principales: l'établissement du contexte, l'identification des risques, l'analyse des risques, l'évaluation des risques et le traitement du risque. Un aperçu du processus de gestion des risques est représenté dans la figure 15. [22]

L'établissement du contexte consiste à définir l'environnement interne et externe d'une organisation dans laquelle les risques doivent être gérés. L'identification des risques se préoccupe de déterminer les événements qui peuvent avoir un impact sur les objectifs de l'organisation, où et quand ces événements peuvent se produire, et pourquoi et comment ils peuvent arriver. L'analyse des risques consiste à développer une compréhension des sources de risques, leurs conséquences positives et négatives et la probabilité que ces conséquences peuvent se produire. L'évaluation des risques se préoccupe de déterminer quels risques justifient le traitement et la priorité relative de chaque traitement. Enfin, le traitement des risques consiste à identifier les options disponibles pour traiter les risques, en évaluant chacune de ces options, et à la préparation et la mise en œuvre des plans de traitement.

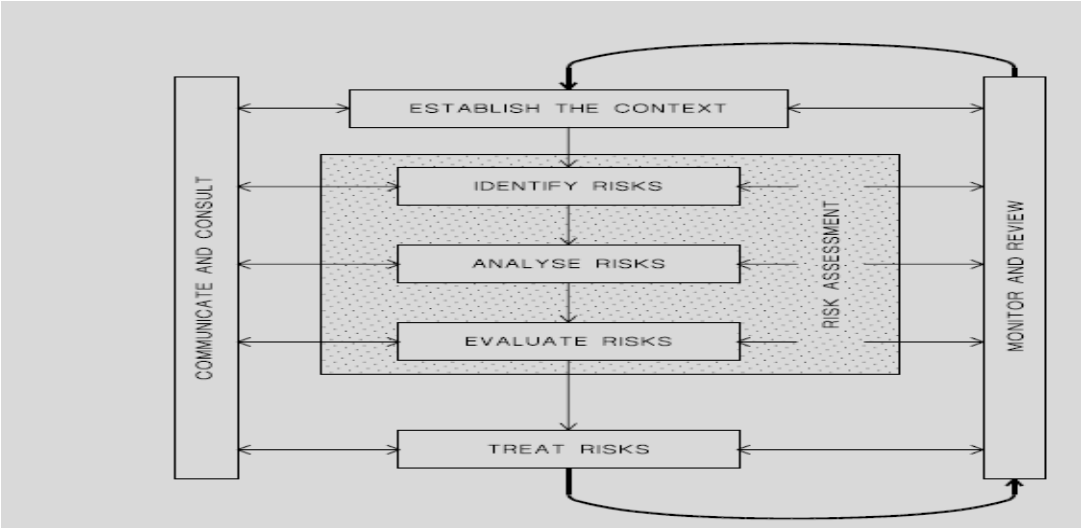


Figure 15: Le processus de gestion de risque

Les modèles traditionnels de contrôle d'accès ne fonctionnent pas bien dans les systèmes informatiques ubiquitaires. Les systèmes informatiques ubiquitaires sont complexes, impliquant des interactions riches entre les diverses entités. Les entités d'un système qui interagissent avec les ressources ne sont pas toujours connues à l'avance. Ainsi, il est presque impossible de construire un périmètre de sécurité bien défini avec lequel le système fonctionne.

Presque tous les modèles traditionnels s'appuient sur l'authentification réussie des utilisateurs prédéfinis, ce qui les rendent inutiles pour les systèmes ubiquitaires. On a plutôt besoin des politiques de sécurité qui utilisent des informations contextuelles.

On proposera dans ce travail un mécanisme de contrôle d'accès basé sur l'évaluation du risque et du contexte. Nous utilisons l'évaluation des risques pour aider le gestionnaire de contrôle d'accès au processus de la prise de décision.

3.3 Objectif de recherche

L'objectif de notre travail est d'identifier les besoins en terme de sécurité des systèmes d'informatique ubiquitaire (chapitre 1), d'exposer l'état de l'art des modèles et politiques de sécurité classiques et de montrer qu'ils s'appliquent mal aux nouvelles exigences (chapitre 2). Et proposer une nouvelle extension du modèle RBAC qui analyse les risques avant de prendre une décision d'accepter ou de refuser l'accès tout en tenant compte des informations du contexte pour déterminer les niveaux de confiance des sujets, le niveau de confiance exigé par chaque rôle et le seuil de risque de l'environnement (chapitre 3).

Pour répondre à ces besoins, nous démontrons comment étendre des modèles traditionnels de contrôle d'accès pour intégrer un raisonnement basé sur le risque et la confiance. En effet, on cherche à élaborer un modèle fort qui répond aux nouvelles exigences des systèmes informatiques. L'évaluation du risque doit se faire de façon dynamique et en temps réel. En outre, on doit minimiser le plus possible les entrées (inputs) dans notre modèle, en proposant des méthodes de calcul qui utilisent des paramètres du contexte.

Pour cela, on va essayer de développer le modèle $RBAC^R$ proposé dans l'article [17], qui est lui-même une extension du modèle RBAC, tout en s'inspirant des idées proposées dans les articles [12], [18] et [20] pour le calcul dynamique, et en tenant compte des informations du contexte, des

valeurs de confiance des utilisateurs. On ajoutera nos propres idées afin de renforcer le plus possible notre modèle surtout pour le calcul du niveau de confiance exigé par le rôle. Pour ce faire, nous introduisons deux tableaux un pour la classification des objets en terme des objectifs de sécurité (confidentialité, intégrité et disponibilité) et le deuxième pour représenter pour chaque action les objectifs de sécurité touchés à l'exécution de l'action. Aussi on proposera notre nouvelle technique de calcul de risque pour l'assignement des rôles. Le calcul se basera sur les scores des règles assignés à un rôle (On détaillera les idées dans le paragraphe « Contribution »).

Récapitulons : Nous cherchons à élaborer un modèle fort de contrôle d'accès qui :

- Sera une extension de RBAC afin de bénéficier des avantages de la notion de rôle : La notion de rôle permet de faciliter l'administration de la politique de sécurité (l'intégration des utilisateurs, la gestion des permissions, la définition de nouveaux objectifs) et de gérer la complexité de gestion des droits d'accès (hiérarchie de rôles).
- N'accordera pas l'accès à un utilisateur en se basant seulement sur son identité mais en élaborant une relation de confiance avec l'utilisateur. Cette relation de confiance sera calculée en se basant sur l'historique, les recommandations et les propriétés de l'utilisateur.
- Calculera le risque à partir de la valeur de confiance de l'utilisateur et ajustera le contrôle d'accès en conséquence.
- Prendra en considération le contexte en intégrant les informations contextuelles dans le calcul de confiance et dans le calcul de risque.
- Prendra en considération la dynamicité de l'environnent en calculant en temps réel les valeurs de confiance et de risque.
- Calculera différemment les valeurs de risque dans les trois étapes de RBAC : affectation des rôles, activation des rôles et exécution des permissions.

3.4 Le nouveau modèle RBAC avec risque

Dans notre modèle on considère que les trois plus importantes étapes dans RBAC sont :

- L'affectation des rôles (Role assignment)
- L'activation des rôles (Role activation)
- L'exécution des permissions (Permission invocation)

Dans RBAC, un utilisateur peut être affecté à un ou plusieurs rôles, et pendant une session l'utilisateur peut activer un ou plusieurs rôles auxquels il est affecté. L'activation des rôles peut se faire selon certaines conditions (des conditions temporelles par exemple : Un utilisateur ne peut activer un certain rôle que de 6 AM à 12 PM). Une fois le rôle activé, l'utilisateur peut exécuter les permissions de ce rôle.

L'estimation de risque se fait différemment selon qu'il s'agit d'affectation des rôles, d'activation des rôles ou d'exécution de permissions. Pour cela notre idée est d'intégrer la notion de risque dans les trois étapes comme montré sur figure17.

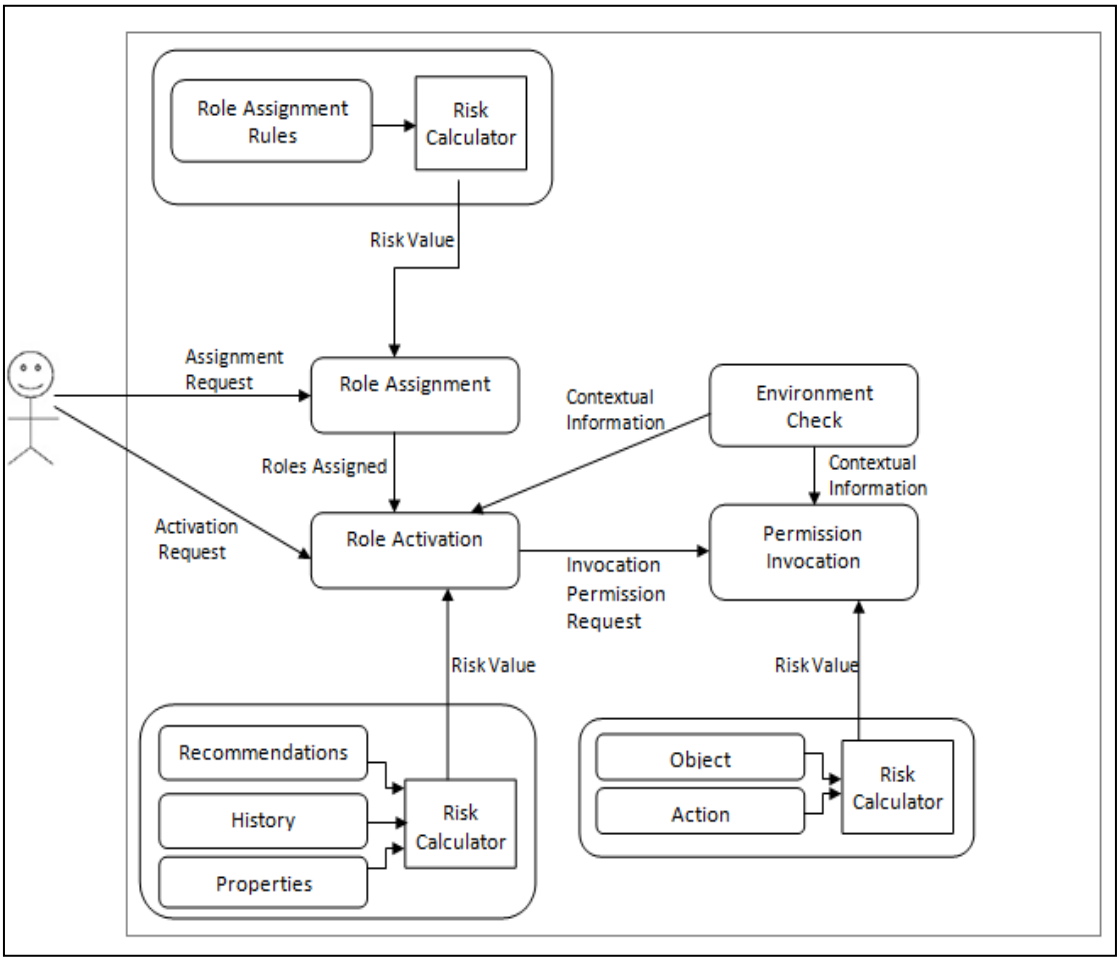


Figure 17: La structure du modèle de contrôle d'accès

Comme montré sur la figure 17, notre système se compose de trois modules : Role Assignment, Role Activation et Permission invocation.

L'utilisateur envoie d'abord une demande d'affectation à un rôle au « Role Assignement ». « Role Assignement » envoie une demande de calcul de risque au « risk calculator». « risk calculator» détermine la valeur de risque en fonction des règles d'assignement et envoie cette valeur au « Role Assignement ». Ce dernier prend la décision en s'appuyant sur cette valeur de risque.

L'utilisateur demande au « Role Activation » d'activer un rôle. « Role Activation » vérifie d'abord si cet utilisateur est affecté à ce rôle. Si oui il demande au « risk calculator » de calculer la valeur de risque. S'il n'est pas affecté à ce rôle la demande est rejetée. « risk calculator » calcule la valeur de risque en fonction des recommandations, de l'historique et des propriétés. Et il envoie cette valeur au « Role Activation ». Ce dernier et en consultant les informations contextuelles détermine le seuil de risque et le compare à la valeur de risque et prendra la décision de permettre l'activation ou rejeter la demande.

Après l'activation, l'utilisateur demande au « Permission Invocation » d'exécuter une permission. « Permission Invocation » demande de calculer la valeur de risque et la décision est prise en fonction de cette valeur calculée et du contexte.

3.4.1 Schéma d'évaluation de risque à l'affectation des rôles aux utilisateurs

Le schéma se compose de sept étapes comme suit :

Étape1 : Identifier, pour chaque rôle, les règles d'affectation au rôle.

Étape2 : Affecter un poids pour chaque règle.

Étape3 : Parmi ces règles, identifier les règles indispensables pour chaque rôle.

Étape4 : Déterminer le niveau de confiance exigé par le rôle qui est la somme des poids des règles indispensables pour le rôle.

Étape5 : Calculer la valeur de confiance de l'utilisateur qui est la somme des poids des règles de l'affectation satisfaites par l'utilisateur.

Étape6 : Déterminer le seuil de risque.

Étape7 : Prendre la décision. Nous avons trois solutions possibles « accepter », « accepter avec risque » ou « refuser ». Nous acceptons la demande d'affectation d'un utilisateur à un rôle lorsque la valeur de confiance de l'utilisateur est supérieure au niveau de confiance exigé par le rôle. Sinon on calcule le risque qui est la différence entre le niveau de confiance exigé par le rôle et la valeur de confiance de l'utilisateur. Nous acceptons avec risque si la valeur de risque est inférieure ou égale à la valeur du seuil de risque sinon nous refusons l'affectation.

La valeur de risque RV_Aff(u, R) est calculée comme suit :

$$\mathrm{RV\_Aff(u,R)} = \begin{cases} 0 & Si \quad CNF\_Aff(u,R) \geq MCA(R) \\ MCA(R) - CNF\_Aff(u,R) & Sinon \end{cases}$$

Avec CNF_Aff(u, R) est le niveau de confiance de l'utilisateur u pour le rôle R à l'étape de l'affectation et MCA(R) est niveau de la confiance exigé par le rôle R à l'étape de l'affectation.

Notons que cette formule est une adaptation de celle représentée dans l'article [17].

Exemple :

Pour chaque affectation à un rôle il y a un ensemble de règles liées à cette affectation et qui doivent être satisfaites par l'utilisateur qui va être affecté à ce rôle.

Dans un rôle, chaque règle est affectée à un score de risque. Si l'utilisateur satisfait toutes les règles alors sa confiance est maximale, sinon sa confiance diminue avec les scores des règles violées.

Le score d'une règle reflète son importance pour le rôle et l'impact de sa violation.

Étape1_Aff : Soit par exemple le rôle x, avec deux règles d'affectation de rôles : (1) l'utilisateur doit avoir l'attribut a, (2) l'utilisateur doit avoir l'attribut b.

Étape2_Aff : Soient 40 et 60 sont, respectivement, le score de violation des règles 1 et 2.

Étape3_Aff : La règle indispensable pour l'affectation au rôle R est d'avoir l'attribut b.

Étape4_Aff : La valeur de confiance exigé par le rôle x est alors 60, c.à.d. seuls les utilisateurs de valeur de confiance CNF_Aff(u, x) supérieure ou égale à 60 peuvent être affectés à x.

Et soient Alice, Bob et Carole trois utilisateurs.

Étape5_Aff : Alice a l'attribut a et b donc sa valeur de confiance pour le rôle x est 100. Bob a seulement l'attribut a, sa valeur de confiance est 40. Carole, comme elle a seulement l'attribut b, sa valeur de confiance est 60.

Étape6_Aff : Soit le seuil de risque = 10.

Étape 7_Aff : D'où Alice et Carole peuvent être affectées à x sans risque contrairement à Bob qui a une valeur de risque RV_Aff(Bob, x) = 20. Cette valeur de risque est supérieur au seuil donc Bob ne peut pas être affecté à x.

3.4.2 Schéma d'évaluation de risque à l'activation des rôles

Le schéma se compose de quatre étapes comme suit :

Étape1_Act : Évaluer une relation de confiance avec l'utilisateur pour le rôle qu'il demande d'activer.

Étape2_Act : Déterminer le niveau de confiance exigé par le rôle pour être activé.

Étape3_Act : Selon les informations contextuelles, déterminer le seuil de risque. Il est important que le seuil de risque soit dynamique et dépend du contexte, puisqu'à chaque fois que les données du contexte changent le seuil doit changer aussi.

Étape4_Act : Prendre la décision. Nous avons trois solutions possibles « accepter », « accepter avec risque » ou « refuser ». Nous acceptons la demande d'activation d'un rôle par un utilisateur lorsque la valeur de confiance de l'utilisateur est supérieure au niveau de confiance exigé par le rôle. Sinon on calcule le risque qui est la différence entre le niveau de confiance exigé par le rôle et la valeur de confiance de l'utilisateur. Nous acceptons avec risque si la valeur de risque est inférieure ou égale à la valeur du seuil de risque sinon nous refusons l'activation.

La valeur de risque RV_Act(u, R) est calculée comme suit :

$$\mathrm{RV\_Act(u,R)} = \begin{cases} 0 & Si \quad CNF\_Act(u,R) \geq MLC(R) \\ MLC(R) - CNF\_Act(u,R) & Sinon \end{cases}$$

Avec CNF_Act(u, R) est le niveau de confiance de l'utilisateur à l'étape d'activation et MLC(R) est le niveau de confiance exigé par le rôle à l'étape d'activation.

Notons que cette formule est une adaptation de celle représentée dans l'article [17].

La décision est prise par cette fonction :

$$Activer(u,R) = \begin{cases} Accepter & RV\_Act(u,R) \leq Seuil\_Risk(\varepsilon,R) \\ refuser & Sinon \end{cases}$$

Nous utilisons $Seuil\_Risk(\varepsilon,R)$ pour spécifier une valeur de seuil pour le risque. Cette valeur est utilisé pour déterminer si le risque est acceptable ou non. Notons qu'on ne peut pas donner une définition générale pour cette fonction. Sa définition est plutôt liée à des applications spécifiques de contrôle d'accès (banque, hôpital, etc.). Par exemple, dans une application bancaire, le seuil peut être plus élevé si les indicateurs économiques sont bons (décrit dans le paramètre ε).

La méthode que nous proposons pour le calcul de niveau de confiance d'un utilisateur à l'étape d'activation des rôles CNF_Act(u,R) est d'élaborer des paramètres desquels dépend le niveau de confiance des utilisateurs.

Nous adoptons le modèle d'évaluation de confiance proposé dans [20] et nous nous concentrons sur trois paramètres pour évaluer la confiance : expérience, propriétés et recommandations.

Dans ce qui suit, nous présentons les méthodes d'évaluation de ses trois paramètres.

Évaluation de l'expérience

L'expérience est modélisée en termes de nombre d'événements rencontrés par un « truster » A concernant un trustee B dans le rôle-contexte c dans un délai de temps spécifié $[t_0, t_n]$. Nous supposons que A a un dossier d'événements depuis t_0. Un événement peut être positif, négatif ou neutre. Les événements positifs contribuent à accroître la composante «croyance (belief)» de l'expérience. Les événements négatifs augmentent la composante «incrédulité (disbelief)» de

l'expérience. Les événements neutres augmentent à la fois et également la composante «croyance» et «incrédulité». Aucune expérience ne contribue à la composante d'«incertitude (uncertainty)» de l'expérience.

Dans la suite, nous décrivons comment calculer l'expérience que le « truster » A a à l'égard du « trustee » B dans le contexte c.

Ceci est formellement notée ${}_AE^c{}_B = (b_E, d_E, u_E)$, où b_E, d_E, u_E représentent, respectivement, les composantes croyance, incrédulité et incertitude, à l'égard de l'expérience que A a envers B.

Soit N l'ensemble des nombres naturels. L'ensemble des instances du temps $\{t_0,t_1,\ldots,t_n\}$est un ensemble totalement ordonné. La relation d'ordre est notée $\prec$, est définie comme suit: $\forall i, j \in \mathrm{N}, t_i \prec t_j \Leftrightarrow i \prec j$.

Nous utilisons le symbole $t_i \preceq t_j$ pour dire soit $t_i \prec t_j$ ou $t_i = t_j$.

Nous utilisons la notation temporelle $[t_i,t_j]$ pour décrire un intervalle de temps où $t_i \preceq t_j$. L'intervalle de temps $[t_i,t_j]$ décrit l'ensemble des instances consécutives du temps où t_i est la première instance et t_j est la dernière.

Nous notons la période de temps d'intérêt $[t_0,t_n]$. Elle est divisée en un ensemble de n sous-intervalles $[t_0,t_1]$, $[t_1,t_2]$, …, $[t_{n-1},t_n]$.

$\forall i, j, k, l \in \mathrm{N}$, avec i, j, k, l toutes distinctes, on a $[t_i, t_j] \cap [t_k, t_l] = \{\ \}$, aussi $\forall i, j, k \in \mathrm{N}$, et i, j, k toutes distinctes $[t_i, t_j] \cap [t_j, t_k] = \{t_j\}$. Toutes les instances, à l'exception de t_0 et t_n, qui se produisent à la limite d'un intervalle, font partie de deux intervalles.

Nous référons à l'intervalle $[t_{k-1}, t_k]$ comme le $K^{ème}$ intervalle, où $0 \le k \le n-1$.

La fonction ET, renvoie l'instant t_j à laquelle un événement donné e_k a eu lieu. Formellement, ET (e_k) = t_j. Par ailleurs, si ET (e_k) = t_j et $t_j \in [t_i, t_k]$ et $j \neq i \wedge j \neq k$, alors on dit que e_k s'est produit dans l'intervalle $[t_i, t_k]$. Pour deux intervalles consécutifs $[t_i, t_j]$ et $[t_j, t_k]$ si ET(e_k) = t_j alors nous supposons que e_k s'est produit dans l'intervalle $[t_i, t_j]$.

On suppose que le fait qu'une expérience est acquise à l'intervalle i, peut être représentée comme (b_i, d_i, u_i) où b_i, d_i, u_i représentent respectivement la croyance, l'incrédulité, et l'incertitude. Si aucun événement ne s'est produit pendant un intervalle de temps particulier i, cela correspond au fait que $u_i = 1$ et $b_i = d_i = 0$.

Le cas suivant est lorsqu'il y a des événements qui se produisent à l'intervalle i.

Soient P_i l'ensemble des événements positifs, Q_i l'ensemble des événements négatifs, et N_i l'ensemble des événements neutres qui se produisent dans l'intervalle i. Chaque événement positif augmente b_i, chaque événement négatif augmente d_i, et chaque événement neutre augmente à la fois b_i et d_i.

Les valeurs de b_i, d_i et u_i sont calculés comme suit :

$$b_i = \frac{|Pi| + \left|\frac{Ni}{2}\right|}{|Pi| + |Ni| + |Qi|}\ ;\ d_i = \frac{|Qi| + \left|\frac{Ni}{2}\right|}{|Pi| + |Ni| + |Qi|}$$ et u_i=0.

Chaque événement positif contribue à la composante croyance par $\frac{1}{|P_i| + |Q_i| + |N_i|}$.

De même, chaque événement négatif contribue à la composante incrédulité par $\frac{1}{|P_i| + |Q_i| + |N_i|}$.

Chaque événement neutre contribue à la fois, également, à la croyance et à l'incrédulité par la composante $\frac{0.5 * |N_i|}{|P_i| + |Q_i| + |N_i|}$.

Par ailleurs, comme il y a des événements qui sont survenus dans l'intervalle, la composante d'incertitude est égale à 0.

Évaluation des propriétés

Un «trustee» communique un ensemble de propriétés physiques qui doivent être vérifiées par le «truster». Des exemples de propriétés pour un périphérique sont la vitesse de traitement d'un

processeur, la capacité de la mémoire, le taux de transmission, la puissance du signal, l'emplacement du capteur, et la sécurité physique. Exemples des propriétés associées à un utilisateur humain sont l'âge, le sexe, le niveau d'éducation, la spécialisation, etc.

L'évaluation du paramètre de propriétés se fait comme suit :

Chaque rôle dans une organisation exige certaines propriétés d'un utilisateur. Les propriétés sont évaluées sur la base des informations fournies par l'utilisateur pour le système lorsqu'il demande l'accès. Chaque rôle R est associé à un ensemble de propriétés positives, PSR = {ps1, ps2,…, psn}, et de propriétés négatives NER = {ne1, ne2,…, nen}. L'ensemble des propriétés positives et négatives sont appelées les propriétés du rôle. Chaque propriété positive et négative est associée à un poids, déterminé par la politique de l'organisation, qui reflète son importance en ce qui concerne le rôle R.

Soit w_{ps1}, w_{ps2},…, w_{psn} les poids des propriétés positives, où $w_{ps_i} \in [0,1], \sum_{i=1}^{n} w_{ps_i} = 1.$

Et soit w_{ne1}, w_{ne2},..., w_{nen} les poids des propriétés négatives, avec $w_{nei} \in [0,1], \sum_{i=1}^{n} w_{ne_i} = 1.$

Soit UP l'ensemble des propriétés possédées par un utilisateur B avec UP = up_1, up_2,..., up_n. Soit $p_B = UP \cap PS_R$ l'ensemble des propriétés positives pour l'utilisateur et qui sont pertinentes pour le rôle, et $n_B = UP \cap NE_R$ l'ensemble des propriétés négatives. Soit w_{psi} le poids de la propriété positive $p_{Bi} \in UP \cap PS_R$, et w_{nei} le poids de la propriété négative $n_{Bi} \in UP \cap NE_R$ et soit $m = |UP \cap PS_R|$, et $n = |UP \cap NE_R|$.

La contribution des propriétés de l'utilisateur à sa confiance est représentée par (b_P, d_P, u_P) où b_P, d_P et u_P désignent, respectivement, la croyance que l'ensemble des propriétés contribuent à l'amélioration de l'opinion sur la fiabilité du «trustee», l'incrédulité que les propriétés peuvent le faire, et l'incertitude.

Chaque $b, d, u \in [0,1]$ et $b + d + u = 1.$

Les valeurs de b_P, d_P et u_P sont calculées en utilisant les formules suivantes:

$$b_p = \frac{\sum_{i=1}^{m} w_{psi}}{\sum_{i=1}^{m} w_{psi} + \sum_{i=1}^{n} w_{nei}};\ d_p = \frac{\sum_{i=1}^{m} w_{nei}}{\sum_{i=1}^{m} w_{psi} + \sum_{i=1}^{n} w_{nei}}$$ et $u_P = 1 - b_P - d_P$.

Évaluation de la recommandation

Nous évaluons la recommandation comme suit :

Le «truster» A peut obtenir une recommandation de multiples recommandeurs concernant le «trustee» B dans le contexte c. L'objectif est de générer un triplet (b, d, u) de chaque recommandeur et de les utiliser pour obtenir (b_R, d_R, u_R) qui représente la recommandation que A a reçu à propos de B par rapport au contexte c. D'abord, on donne des précisions sur la façon dont le triplet est calculé pour chaque recommandeur. Plus tard, on décrit comment ces résultats sont agrégés.

Soit M un recommandeur. Le recommandeur M peut ou non avoir une relation de confiance avec le «trustee» B dans le contexte c. Le «truster» A peut fournir un questionnaire au recommandeur. Le recommandeur est autorisé à utiliser les valeurs 1, -1, 0 ou ⊥ en remplissant ce questionnaire. La valeur 1 indique la croyance, -1 indique l'incrédulité, 0 indique neutre, et ⊥ indique inconnu. Le nombre de ⊥s par rapport au nombre total des valeurs va donner la mesure de l'incertitude (uncertainty). Le nombre de 1s avec la moitié du nombre de 0s par rapport au nombre total des valeurs donne la valeur de la croyance (belief). Le nombre de -1s avec la moitié du nombre de 0s par rapport au nombre total des valeurs donne la valeur de l'incrédulité (disbelief).

Si le recommandeur ne retourne pas une recommandation, le «truster» utilise le triplet (0,0,1) comme une recommandation de M.

Le « truster » A aura une relation de confiance avec le recommandeur M. Le contexte de cette relation de confiance sera d' «agir de manière fiable pour fournir un service (recommandation, dans ce cas)».

Cette relation de confiance aura une influence sur l'opinion de la recommandation prévue par le recommandeur.

Le «truster» évalue l'opinion du recommandeur à propos du «trustee» avec cette valeur de confiance. L'évaluation de la recommandation basée sur la relation de confiance entre le «truster»

et le recommandeur a un avantage important. Supposons que le recommandeur raconte un mensonge à propos du «trustee» dans sa recommandation en vue d'obtenir un avantage avec le «truster».

La relation de confiance qu'a le «truster» A avec le «trustee» M dans le contexte de la fourniture d'une recommandation est représentée comme une matrice 3 * 3. Les lignes de la matrice correspondent à l'expérience, les connaissances et la recommandation et les colonnes correspondent à la croyance (belief), l'incrédulité (disbelief), et l'incertitude (uncertainty).

Cette matrice est normalisée comme sera décrit dans le paragraphe «Normalisation du vecteur de la confiance» et converti en un triplet de la forme (b, d, u). Ce triplet sera utilisé pour l'évaluation de l'opération.

Si le recommandeur M ne croit pas le «trustee» B ou il est incertain à propos de B, alors A ne croit pas aussi B ou il est incertain à propos de B. En outre, l'incrédulité et l'incertitude de A quant à l'opinion de M contribuent à l'incertitude de A envers B. Si M envoie un triplet ($_Mb_B$, $_Md_B$, $_Mu_B$) comme une recommandation de B, et A a confiance en M comme ($_Ab_M$, $_Ad_M$, $_Au_M$), alors la recommandation $_{AM}R^c_B$ du recommandeur M de B pour le « truster » A dans un contexte c est donnée par ($_{AM}b^c_B$, $_{AM}d^c_B$, $_{AM}u^c_B$). Les valeurs de $_{AM}b^c_B$, $_{AM}d^c_B$ et $_{AM}u^c_B$ sont calculées comme suit :

$$_{AM}b^c_B = {}_Ab_M \times {}_Mb_B,\ {}_{AM}d^c_B = {}_Ad_M \times {}_Md_B,\ {}_{AM}u^c_B = {}_Ad_M + {}_Au_M + {}_Ab_M \times {}_Mu_B.$$

Rappelons que le « truster » A peut obtenir des recommandations pour le « trustee » B de nombreux recommandeurs différents. Alors la croyance de A pour une recommandation à propos de B est la moyenne des valeurs de croyance de toutes les recommandations et l'incrédulité de A est la moyenne des valeurs de l'incrédulité de toutes les recommandations. La même chose pour l'incertitude de A pour les recommandations. Par conséquent, si Ψ est un groupe de n recommandeurs alors

$$_{A\Psi}b_R = \frac{\sum_{i=1}^{n} {}_{Ai}b^c_B}{n},\ {}_{A\Psi}d_R = \frac{\sum_{i=1}^{n} {}_{Ai}d^c_B}{n},\ {}_{A\Psi}u_R = \frac{\sum_{i=1}^{n} {}_{Ai}u^c_B}{n}.$$

Par conséquent, la composante recommandation est exprimée par le triplet :

(${}_{A\Psi}b_R$, ${}_{A\Psi}d_R$, ${}_{A\Psi}u_R$).

Normalisation du vecteur de la confiance

Après avoir déterminé les triplets pour chaque composante de la confiance, nous spécifions la relation de confiance entre le «truster» A et le «trustee» B dans un rôle-contexte c au temps t comme suit :

$$\left(A \xrightarrow{c} B\right)_t = \begin{pmatrix} b_P & d_P & u_P \\ b_E & d_E & u_E \\ {}_{A\Psi}b_R & {}_{A\Psi}d_R & {}_{A\Psi}u_R \end{pmatrix}$$

Deux «trusters» peuvent avoir deux valeurs de confiance différentes pour le même «trustee». Cela peut se produire car un «truster» peut assigner des poids différents aux différents facteurs qui influent la confiance. Un «truster» peut donner plus de poids à l'un des paramètres dans le calcul d'une relation de confiance. Par exemple, un «truster» A peut choisir de donner plus d'importance à l'expérience qu'à la recommandation dans le calcul de la confiance. Quel élément doit avoir plus d'importance que les autres, est une question de la politique d'évaluation de la confiance du «truster».

La politique est représentée par le «truster» comme un vecteur de la politique de confiance. Les éléments de ce vecteur sont les poids correspondant aux paramètres de la relation de la confiance.

Soit $\left(A \xrightarrow{c} B\right)_t$ la relation simple de confiance entre A et B dans le contexte c au temps t. Soit également ${}_AW^c{}_B = [W_P, W_E, W_R]$ le vecteur de la politique d'évaluation de la confiance tel que $W_P + W_E + W_R = 1$ et $W_P, W_E, W_R \in [0,1]$.

Par conséquent, la relation de confiance normalisée entre un « truster » A et un « trustee » B à un instant t et pour un contexte c est donné par :

$$\left(A \xrightarrow{c} B\right)_t^N = {}_AW^c{}_B \times \left(A \xrightarrow{c} B\right)_t$$

$$= (W_E, W_k, W_R) \times \begin{pmatrix} b_P & d_P & u_P \\ b_E & d_E & u_E \\ {}_{A\Psi} b_R & {}_{A\Psi} d_R & {}_{A\Psi} u_R \end{pmatrix}$$

$$= ({}_A\hat{b}^c{}_B, {}_A\hat{d}^c{}_B, {}_A\hat{u}^c{}_B)$$

Avec ${}_A\hat{b}^c{}_B = W_P \times b_P + W_E \times b_E + W_R \times {}_{A\Psi} b_R$,

${}_A\hat{d}^c{}_B = W_P \times d_P + W_E \times d_E + W_R \times {}_{A\Psi} d_R$,

${}_A\hat{u}^c{}_B = W_P \times u_P + W_E \times u_E + W_R \times {}_{A\Psi} u_R$,

${}_A\hat{b}^c{}_B$, ${}_A\hat{d}^c{}_B$, ${}_A\hat{u}^c{}_B \in [0,1]$ et

${}_A\hat{b}^c{}_B + {}_A\hat{d}^c + {}_A\hat{u}^c{}_B = 1.$

Après l'évaluation des trois paramètres sur lesquels se base la confiance, on obtient la valeur de confiance qui est calculée comme suit :

$$T = \frac{{}_A\hat{b}^c{}_B + {}_A\hat{u}^c{}_B}{{}_A\hat{b}^c{}_B + {}_A\hat{d}^c{}_B + {}_A\hat{u}^c{}_B}$$

La valeur de T sera de l'ordre [0,1]. La valeur proche de 0 indique une faible valeur de confiance de l'utilisateur B en ce qui concerne le rôle c, tandis que la valeur proche de 1 indique une valeur de confiance très élevée de l'utilisateur en ce qui concerne le rôle c.

En effet, en suivant les étapes de calcul de confiance de ce modèle, on arrivera à fixer les paramètres du contexte desquels dépend la confiance, et qui sont les propriétés, l'expérience et les recommandations, ensuite on calcule pour l'utilisateur u ,qui demande l'activation d'un rôle, les triplets pour chaque paramètre de cette forme (b,d,u), puis à partir de ces triplets on arrivera à calculer le vecteur de confiance et à partir de ce vecteur on déterminera la valeur de confiance.

Exemple :

Étape1_Act : Calcul de niveau de confiance d'un utilisateur à l'étape d'activation des rôles CNF_Act(u,R).

Évaluation des propriétés :

Soit R un rôle qui exige quatre propriétés pour être activé : $\{P_1, P_2, P_3, P_4\}$. Les poids de ces propriétés sont respectivement $W_{p1} = 30$, $W_{p2} = 40$, $W_{p3} = 10$, $W_{p4} = 20$.

L'utilisateur u qui demande l'activation de R satisfait seulement P_1 et P_2.

Les valeurs de b_p, d_p et u_p sont alors calculées comme suit :

$$b_p = \frac{30+40}{100} = 0.7, d_p = \frac{20+10}{100} = 0.3, u_p = 0.$$

Évaluation de l'expérience :

L'expérience est quantifiée sur la base des événements performés par l'utilisateur u pour le rôle R au cours des cinq dernières années et l'unité de la période du temps est égale à un an. Le poids de chaque intervalle de temps (W_{slot_i}) représente la période la plus proche de l'heure actuelle est définie par la politique comme suit: $W_{slot\_1} = \frac{5}{15}$, $W_{slot\_2} = \frac{4}{15}$, $W_{slot\_3} = \frac{3}{15}$, $W_{slot\_4} = \frac{2}{15}$, $W_{slot\_5} = \frac{1}{15}$.

Pendant la 1ère année (5ème période), l'utilisateur u a fait quatre événements positifs et un événement négatif.

$$b_{slot\_5} = \frac{4}{5} = 0.8,\ d_{slot\_5} = \frac{1}{5} = 0.2,\ u_{slot\_5} = 0.$$

Pendant la 2ème année (4ème période), u a fait deux événements positifs, trois événements négatifs.

$$b_{slot\_4} = \frac{2}{5} = 0.4,\ d_{slot\_4} = \frac{3}{5} = 0.6,\ u_{slot\_4} = 0.$$

Pendant la 3ème année, u n'a pas activé son rôle de chirurgien.

$b_{slot\_3}= 0$, $d_{slot\_3}= 0$, $u_{slot\_3}= 1$.

Pendant la 4ème année, u a fait trois événements positifs, et deux événements neutres.

$b_{slot\_2} = \frac{3+\frac{2}{2}}{5} = 0.8$, $d_{slot\_2} = \frac{0+\frac{2}{2}}{5} = 0.2$, $u_{slot\_2} = 0$.

Pendant la 5ème année, u a fait un événement positif, deux événements négatifs, et deux événements neutres.

$b_{slot\_1} = \frac{1+\frac{2}{2}}{5} = 0.4$, $d_{slot\_1} = \frac{2+\frac{2}{2}}{5} = 0.6$, $u_{slot\_1} = 0$.

Les b_E, d_E et u_E seront alors calculés comme suit:

$$b_E = \frac{5}{15}\times 0.4 + \frac{4}{15}\times 0.8 + \frac{3}{15}\times 0 + \frac{2}{15}\times 0.4 + \frac{1}{15}\times 0.8 = 0{,}35$$

$$d_E = \frac{5}{15}\times 0.6 + \frac{4}{15}\times 0.2 + \frac{3}{15}\times 0 + \frac{2}{15}\times 0.6 + \frac{1}{15}\times 0.2 = 0{,}45$$

$$u_E = \frac{5}{15}\times 0 + \frac{4}{15}\times 0 + \frac{3}{15}\times 1 + \frac{2}{15}\times 0 + \frac{1}{15}\times 0 = 0,\ 2$$

Évaluation des recommandations :

Soient deux recommandeurs M1 et M2. La relation de confiance avec M1 est (0.96, 0.02, 0.02), et la relation de confiance avec M2 est (0.9, 0.05, 0.05).

La recommandation de M1 pour l'utilisateur u est (0.98, 0.00, 0.02) et la recommandation de M2 est (0.99, 0.00, 0.01).

La recommandation de M1 est calculée comme suit :

$_{M1}b_R = 0.96\times 0.98 = 0.94$, $_{M1}d_R = 0.02\times 0.00 = 0.00$, $_{M1}u_R = 0.02 + 0.02 + 0.96\times 0.02 = 0.06$

.

La recommandation de M2 est calculée comme suit :

$_{M2}b_R = 0.90\times 0.99 = 0.89$, $_{M2}d_R = 0.05\times 0.00 = 0.00$, $_{M2}u_R = 0.05 + 0.05 + 0.9\times 0.01 = 0.11$.

En utilisant ces informations on détermine les valeurs de b_R, d_R et u_R :

$b_R = \frac{0.94+0.89}{2} = 0.92$, $d_R = \frac{0.00+0.00}{2} = 0.00$, $u_R = \frac{0.06+0.11}{2} = 0.08$.

La politique du système S donne plus d'importance à l'expérience (W_E= 0.76) qu'aux propriétés (W_P= 0.12) et recommandations (W_R= 0.12).

Le vecteur de la confiance est alors :

$$\left(H \xrightarrow{R} u\right) = ({}_S b_u^R, {}_S d_u^R, {}_S u_u^R) = (W_P, W_E, W_R) \times \begin{pmatrix} b_p & d_p & u_p \\ b_E & d_E & u_E \\ b_R & d_R & u_R \end{pmatrix} = (0.76, 0.12, 0.12) \times \begin{pmatrix} 0.7 & 0.3 & 0 \\ 0.35 & 0.45 & 0.2 \\ 0.92 & 0 & 0.08 \end{pmatrix}$$

$$= (0.68, 0.28, 0.04)$$

Et la valeur de confiance est : CNF_Act(u, R)= $\frac{0.68+0.04}{1} = 0.72$.

Étape2_Act : Calcul de niveau de confiance exigé par le rôle MLC(R)

Dans notre système, chaque rôle exige un niveau de confiance qu'on doit comparer avec le niveau de confiance d'un utilisateur lors de l'opération de l'activation de rôle.

Étant donné qu'un rôle est un ensemble de permissions, pour calculer le niveau de confiance exigé d'un rôle il faut calculer pour chaque permission de ce rôle son niveau de sensibilité. Et le niveau de confiance du rôle est le niveau de sensibilité le plus élevée de ses permissions.

Soit le rôle R ={P1, P2,…,Pn}, et soient LS(P1), LS(P2),…, LS(Pn) les niveaux de sensibilité des permissions de R.

MLC(R) = Max (LS(P1), LS(P2),…, LS(Pn)).

Notre méthode de calcul du niveau de sensibilité d'une permission LS(P) est la suivante :

Dans notre système on introduit cette table de classification de données où pour chaque objet on précise le niveau de confidentialité, d'intégrité et de disponibilité.

	Confidentialité	Intégrité	Disponibilité
O1	CO1	IO1	DO1
O2	CO2	IO2	DO2
…	…	…	…
On	Con	Ion	Don

Table 6 : Classification des objets

Ici COi, IOi et DOi représentent respectivement le niveau de confidentialité, le niveau d'intégrité et le niveau de disponibilité demandés pour l'objet Oi.

Dans une autre table, pour chaque action on précise les objectifs de sécurité qui peuvent être touchés une fois que cette action est exécutée.

Action	Objectifs de sécurité
READ(O)	CONFIDENTIALITÉ
APPEND(O)	INTÉGRITÉ
WRITE(O)	INTÉGRITÉ, DISPONIBILITÉ
MODIFY(O)	CONFIDENTILITÉ, INTÉGRITÉ, DISPONIBILITÉ
DELETE(O)	DISPONIBILITÉ
…	…

Table 7 : Les objectifs de sécurité affectés par les actions

Dans cette table on suppose que WRITE(O) = {APPEND(O), DELETE(O)} et MODIFY(O) = {READ(O), DELETE(O), APPEND(O)}.

À partir de ces deux tables on détermine le niveau de sensibilité des permissions.

Soit P(O, A) une permission qui permet d'exécuter l'action A sur l'objet O. On retire de la table7 les objectifs de sécurité menacés à l'exécution de A. Pour chaque objectif tiré de la table 7 on détermine à partir de la table6 le niveau de O en termes de cet objectif.

Par exemple soit la permission P(O2, WRITE). Le niveau de sensibilité de cette permission se calcule en suivant ces étapes :

Étape 1 : On retire de la table 7 les objectifs de sécurité menacés à l'exécution de WRITE.

Soit Extract-objectives(Action) la fonction qui retourne les objectifs menacés à l'exécution d'une action « Action ».

Action	Objectifs de sécurité
READ(O)	CONFIDENTIALITÉ
APPEND(O)	INTÉGRITÉ
WRITE(O)	INTÉGRITÉ, DISPONIBILITÉ
MODIFY(O)	CONFIDENTILITÉ, INTÉGRITÉ, DISPONIBILITÉ
DELETE(O)	DISPONIBILITÉ
…	…

Figure 16: Extraction d'objectifs affectés à l'exécution de WRITE

Comme le montre la figure 16, Extract-objectives(WRITE) = {Intégrité, Disponibilité}.

Étape 2 : On retire de la table 6 le niveau de O2 en termes des objectifs menacés {Intégrité, Disponibilité}.

Soit Classification (Object, Objective) la fonction qui retourne le niveau de l'objet « Objet » et termes de l'objectif « Objective ».

	Confidentialité	Intégrité	Disponibilité
O1	CO1	IO1	DO1
O2	CO2	IO2	DO2
…	…	…	…
On	Con	Ion	Don

Figure 17: Classification de O2 en termes de : Intégrité et Disponibilité

Comme le montre la figure 17, Classification (O2, Intégrité) = IO2 et Classification (O2, Disponibilité) = DO2

Étape 3 : Le niveau de sensibilité d'une permission P(A,O) est déduit des niveaux des objectifs de sécurité de O menacés à l'exécution de A.

On peut par exemple considérer le niveau maximal des niveaux des objectifs de sécurité de O menacés à l'exécution de A comme le niveau de sensibilité de la permission P(O2, write).

Dans notre exemple : LS (O2, write) = Max(Classification (O2, Intégrité), Classification (O2, Disponibilité)).

On applique ce principe sur toutes les permissions du rôle et le niveau de confiance exigé pour ce rôle sera le niveau de sensibilité le plus élevé des permissions de ce rôle.

MLC(R) = Max(LS(P1), LS(P2),…, LS(Pn)).

Étape3_Act : Déterminer un seuil de risque. Le seuil est dynamique et ajusté par rapport aux informations contextuelles du temps actuel.

Par exemple, dans une situation normale l'administrateur décide de ne pas accepter les risques et donc il met le seuil à zéro et seuls les utilisateurs de valeurs de confiance CNF_Act(u, R) supérieures ou égales à MLC(R) peuvent activer R.

Étape4_Act : La prise de décision.

3.4.3 Le schéma d'évaluation de risque à l'exécution des permissions

Le schéma se compose de deux étapes comme suit :

Étape1_Exe : affecter des valeurs de risk_acceptance à chaque permission d'un rôle.

Étape2_Exe : Prendre la décision. La permission peut être exécutée si la valeur de confiance de l'utilisateur est supérieure au niveau de sensibilité de la permission. Sinon nous soustrayons la valeur de risk_acceptance de la valeur du niveau de sensibilité de la permission. Nous acceptons l'exécution si la valeur de confiance de l'utilisateur est supérieure au résultat de la soustraction sinon nous refusons la demande de l'exécution de la permission.

La décision est prise par la formule suivante :

$$Executer(U,P) = \begin{cases} Accepter & Si \quad CNF\_Act(U,R) \geq LS(P) - risk\_acceptance(P) \\ \text{Re } fuser & Sinon \end{cases}$$

Exemple :

Soit le rôle R avec P_R l'ensemble des permissions affecté à R et soit P_R= {P_1, P_2, P_3}. Soient $LS(P_1) = 30$, $LS(P_2) = 50$, $LS(P_3) = 25$ les niveaux de sensibilité de ces permissions.

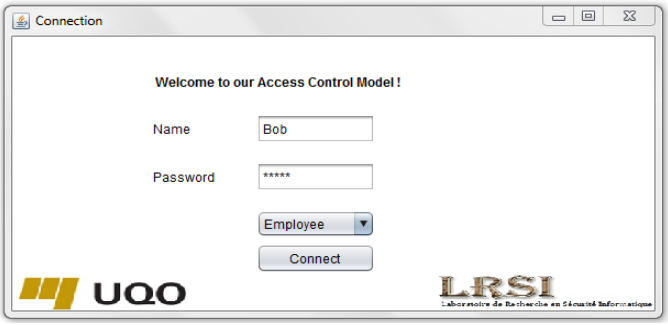
Étape1_Exe : Soient $risk\_acceptance(P_1) = 2$, $risk\_acceptance(P_2) = 5$, $risk\_acceptance(P_3) = 4$ les valeurs acceptées de risques pour ces permissions.

Soit la valeur de confiance de l'utilisateur CNF_Act(u, R)= 35. On suppose que la demande d'activation du rôle R par l'utilisateur u a été acceptée.

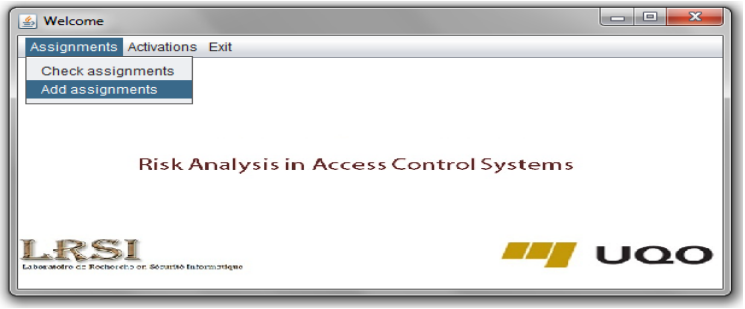
Étape2_Exe : L'utilisateur u demande d'exécuter P1. Cette demande est acceptée (CNF_Act(u,R) > LS(P1)). u demande l'exécution de P2. Mais cette demande est refusée (CNF_Act(u,R) < LS(P1) - risk_acceptance(P2)). u demande d'exécuter P3 et cette demande est acceptée (CNF_Act(u,R) > LS(P3)).

3.5 L'outil

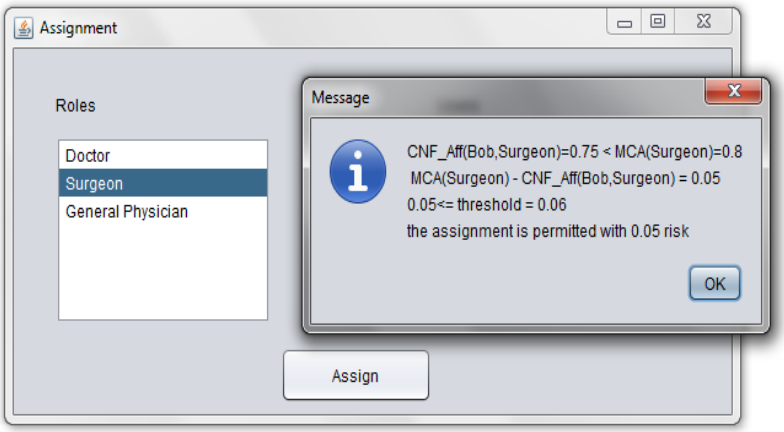
Dans cette section, nous présentons l'outil que nous avons développé pour créer un nouveau modèle de contrôle d'accès basé sur l'estimation de risque.



La figure suivante représente l'interface de l'outil développé, où on demande à l'utilisateur qui souhaite se connecter à notre système de taper son login et mot de passe.

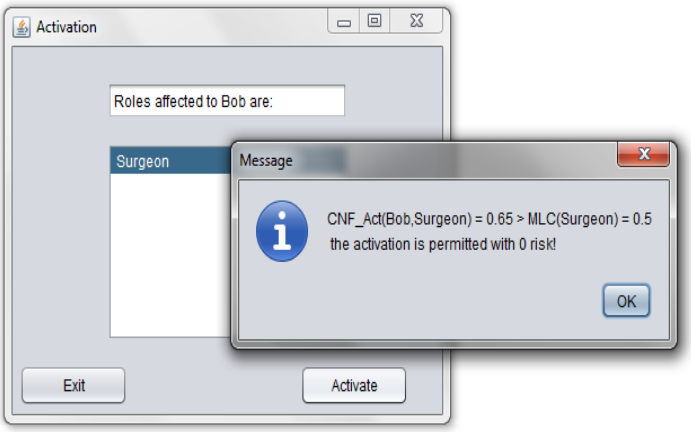


L’utilisateur Bob est maintenant connecté à notre système. Il demande d’affectation.



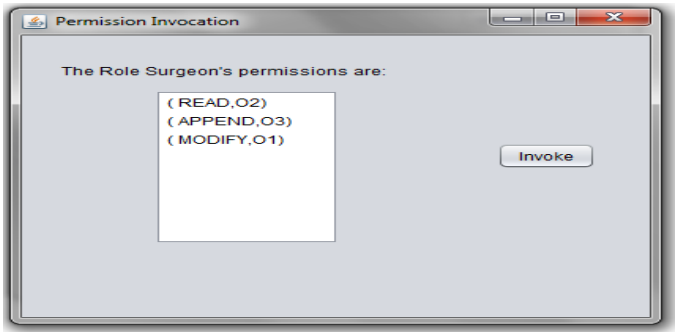
L’utilisateur demande l’affectation au rôle Surgeon.

Et la demande d’affectation a été permise avec 0.05 de risque.



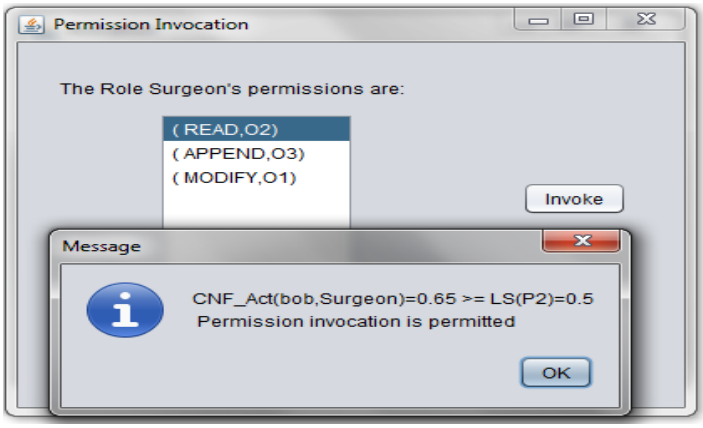
Une fois affecté au rôle surgeon, Bob demande de l’activer.

Et l’activation est permise sans risque.

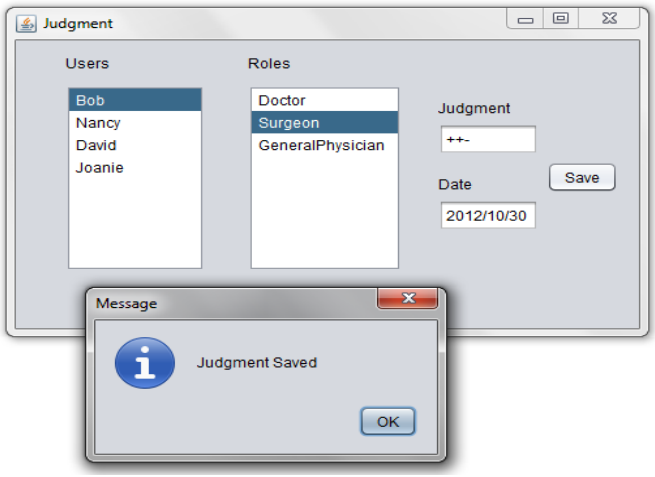


Le rôle Surgeon possède trois permissions.

On demande à Bob de choisir quelle permission il souhaite exécuter.



Bob choisit de lire l'objet O2. Cette permission est permise.



Après chaque activation d'un rôle, l'administrateur juge la performance de l'utilisateur. On voit dans cette fenêtre que l'administrateur a jugé que Bob en tant que Surgeon a fait 2 événements positifs et un événement négatif. Ce jugement sera enregistré et influencera la nouvelle valeur de confiance de Bob.

3.6 Évaluation

En revenant à notre liste de besoins qu'on a dressé au paragraphe 3.3, on trouve qu'on a été fidèle à notre liste et qu'on a répondu à tous les points.

En effet, notre modèle :

- Est une extension de RBAC
- N’accorde pas l’accès à un utilisateur en se basant seulement sur son identité mais en élaborant une relation de confiance avec l’utilisateur.
- Calcule le risque à partir de la valeur de confiance de l’utilisateur et ajuste le contrôle d’accès en conséquence.
- Prend en considération le contexte.
- Prend en considération la dynamicité de l’environnent en calculant en temps réel les valeurs de confiance et de risque.
- Calcule différemment les valeurs de risque dans les trois étapes de RBAC.

3.7 Conclusion

Ce travail nous a permis de nous pencher sur un sujet important concernant les systèmes d’information. En effet, nous nous sommes intéressés au contrôle d’accès pour les nouveaux systèmes informatiques, les systèmes ubiquitaires.

Pour atteindre un niveau de protection satisfaisant pour les systèmes informatiques ubiquitaires, il convient de définir un modèle de contrôle d’accès correspondant aux besoins. Un modèle de contrôle d’accès "ubiquitaire" doit permettre une administration dynamique des ressources par leur propriétaire. Pour cela on a proposé un nouveau modèle extensible de RBAC et qui se base sur l’estimation de risque.

Dans notre modèle, les décisions sont prises dynamiquement et en se basant, explicitement, sur le risque. On propose trois méthodes de prise de décision : une méthode à l’étape des affectations des rôles, une deuxième à l’étape de l’activation des rôles et une troisième à l’étape de l’exécution des permissions.

Le travail ci-dessus est encore à l'état embryonnaire. Dans les travaux futurs, nous devons tenir compte d'autres paramètres et facteurs qui affectent le processus d'évaluation du risque et essayer de trouver une solution pour le calcul dynamique du seuil de risque de l’environnement qui reste la faiblesse de tous les modèles de contrôle d’accès proposés de nos jours.

Chapitre 4 : Conclusion

La notion de modèle de contrôle d'accès est apparue avec la multiplication des systèmes informatiques et des enjeux de leur sécurité. L'objectif de tout contrôle d'accès est de permettre de décider si une demande d'accès est légitime ou non. Cette décision est prise au regard d'une politique organisée selon un modèle.

Dans les nouveaux systèmes informatiques, les systèmes ubiquitaires, les entités fonctionnent dans des environnements qui sont dynamiques et imprévisibles, les obligeant à être capables de faire face à des interactions inattendues et des utilisateurs inconnus à l'avance.

Les modèles traditionnels de contrôle d'accès sont souvent non appropriés pour des applications de l'informatique ubiquitaire.

En effet, on a besoin des nouveaux modèles de contrôle d'accès qui ne considèrent pas seulement les informations contextuelles, mais évaluent aussi l'effet de cette information sur le niveau du risque global du système.

Lorsqu'un système accorde des privilèges à un utilisateur, c'est avec l'espoir qu'ils seront utilisés d'une manière particulière, mais il y a aussi la possibilité que l'utilisateur s'écarte du comportement attendu. La probabilité et la gravité combinées de cette variation est le risque d'accorder ces privilèges à l'utilisateur.

4.1 Travail accompli

Le modèle de risque proposé, est une nouvelle extension de RBAC. Avant de pouvoir accéder à une ressource critique un utilisateur doit passer par trois étapes importantes:

- L'affectation des rôles : c'est l'étape des affectations de l'utilisateur aux rôles. L'affectation est permise si l'utilisateur répond à un minimum exigé des règles d'affectation. La valeur de risque est quantifiée dans le cas contraire, et l'affectation n'est permise que si le risque ne dépasse pas un certain seuil.

- L’activation des rôles : à cette étape une relation de confiance est établie entre l’utilisateur et le système et l’activation n’est permise que lorsque la confiance est supérieure ou égale à la confiance exigée par le rôle. Si cette condition n’est pas vérifiée, on passe à l’évaluation de risque. L’activation n’est permise que si le risque est inférieur ou égal à un certain seuil de risque.
- L’exécution de la permission : à cette étape on décide de permettre l’exécution de la permission demandée ou non. A chaque permission on lui accorde une valeur acceptée de risque. L’exécution n’est permise que lorsque la confiance à l’utilisateur est supérieure ou égale au niveau de sensibilité de la permission moins la valeur de risque acceptée.

Les avantages de notre modèle c’est qu’il est plus flexible et plus adapté aux exigences des environnements dynamiques que les autres modèles qu’on a présenté dans notre recherche bibliographique. Pour chaque étape de RBAC un modèle de gestion de risque est établi. La quantification des valeurs de risque et de la confiance se font dynamiquement et en temps réel en tenant compte des informations contextuelles.

4.2 Travaux futurs

Bien que nous ayons accompli un certain nombre de contributions mentionnées dans la section précédente, d’autres améliorations sont encore possibles. La suite de ce travail pourrait être l’extension de la méthode présentée pour qu’elle prenne en compte d’autres paramètres pour le calcul de la confiance et une fonction pour le calcul dynamique du seuil du risque.

Dans notre travail nous n'avons pas considéré les règles de délégation Il serait intéressant d'étendre notre système pour les prendre en considération.

Bibliographie

[1] B. W Lampson, «ACM SIGOPS Operating Systems,» *The Special Interest Group on Operating Systems (SIGOPS),* p. 18–24, 1974.

[2] R. THION, «Thèse : Structuration Relationnelle Des Politiques De Contrôle D'accès Représentation, Raisonnement et vérification logiques,» L'Institut National des Sciences Appliquées de Lyon, Lyon, 2008.

[3] M. Weiser, «The computer for the 21st century,» *Pervasive Computing, IEEE,* vol. 6, pp. 19 - 25, Jan.-March 2002.

[4] H. Wang, Y. Zhang et C. Jinli, «Ubiquitous Computing Environments and Its Usage Access Control,» *Proc. of the First International Conference on Scalable Information Systems,* 2006.

[5] R. S. Sandhu et P. Samarati, «Access Control : Principles and Practice,» *Communications Magazine, IEEE,* vol. 9, pp. 40 - 48, September 1994.

[6] A. Abou El Kalam et Y. Deswarte, «Contrôle d'accès basés sur les rôles, les groupes d'objets et le contexte : Étude de cas dans les Systèmes d'information et de Communication en Santé,» LAAS-CNRS, Toulouse, 2006.

[7] M. J. Moyer et M. Ahamad, «Generalized Role-Based Access Control,» *Distributed Computing Systems, 2001. 21st International Conference on,* pp. 391 - 398, 2001.

[8] G. Zhang et M. Parashar, «Context-aware Dynamic Access Control for Pervasive Applications,» *Proc. of the Communication Networks and Distributed Systems Modeling and Simulation Conference,* pp. 21-30, 2004.

[9] F. Feng, C. Lin, D. Peng et J. Li, «A Trust and Context Based Access Control Model for Distributed Systems,» *Proc. of the 2008 10th IEEE International Conference on High Performance Computing and Communications,* pp. 629-634, 2008.

[10] J. Wook Woo, M. Jin Hwang, C. Gyeong Lee et H. Yong Youn, «Dynamic Role-Based Access Control with Trust-Satisfaction and Reputation for Multi-agent System,» *Proc. of IEEE 24th International Conference on Advanced Information Networking and Applications Workshops,* pp. 1121-1126, 2010.

[11] N. Dimmock, «How much is "enough"? Risk in trust-based access control,» pp. 281 - 282 , June 2003 .

[12] H. Le Xuan, Z. Yonil, L. Sungyoung, L. Young-Koo et L. Heejo, «Enforcing Access Control Using Risk Assessment,» *Universal Multiservice Networks,* vol. 5, pp. 419- 424, 2007.

[13] A. Ali et Z. Ning, «A Context-Risk-Aware Access Control model for Ubiquitous environments,» *Computer Science and Information Technology,* vol. 7, pp. 775-782 , 2008.

[14] A. Sundaram, «An introduction to intrusion detection,» *Crossroads - Special issue on computer security,* pp. 3-7 , March 1996 .

[15] F. H. Barron, «Selecting a best multiattribute alternative with partial information about attribute weights,» *cta Psychologica,* pp. 91-103, August 1992.

[16] R. A. Shaikh, K. Adi, L. Logrippo et S. Mankovski, «Risk-based Decision Method for Access Control Systems,» *Proc. of Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on,* pp. 189 - 192, 2011.

[17] J. Ma, K. Adi, M. Mejri et L. Logrippo, «Risk Analysis in Access Control Systems,» *Proc. of eighth Annual International Conference on Privacy, Security and Trust,* pp. 160-166, 2010.

[18] I. Ray, I. Ray et S. Chakraborty, «An interoperable context sensitive model of trust,» *Journal of Intelligent Information Systems,* pp. 75-104, 2009.

[19] A. Jøsang, «Artificial Reasoning with Subjective Logic,» *Proceedings of the 2nd Australian Workshop on Commonsense Reasoning,* 1997.

[20] M. Toahchoodee, R. Abdunabi, I. Ray et I. Ray, «A Trust-Based Access Control Model for Pervasive Computing Applications,» *IFIP International Federation for Information Processing,* pp. 307-314, 2009.

[21] A. Ahmed et N. Zhang, «A Context-Risk-Aware Access Control Model for Ubiquitous Environments,» *Proc. of 3rd International Workshop on Secure Information Systems (SIS'08), IEEE CS Press,* p. 775–782, 2008.

[22] Casualty Actuarial Society, «Entreprise Risk Management Committee,» vol. 62, Mai 2003.