A model to reduce complexity and maintain coherence between Access Control and Transmission Control policies

Yoann Bertrand, Mireille Blay-Fornarino, Karima Boudaoud, Michel Riveill

LABORATOIRE

INFORMATIQUE, SIGNAUX ET SYSTÈMES DE SOPHIA ANTIPOLIS
UMR 7271

A model to reduce complexity and maintain coherence between Access Control and Transmission Control policies

Yoann Bertrand, Mireille Blay-Fornarino, Karima Boudaoud, Michel Riveill
EQUIPE SPARKS

Rapport de Recherche

Mai 2016

Laboratoire d'Informatique, Signaux et Systèmes de Sophia-Antipolis (I3S) - UMR7271 - UNS CNRS
2000, route des Lucioles — Les Algorithmes - bât. Euclide B — 06900 Sophia Antipolis — France
http ://www.i3s.unice.fr

Membre de UNIVERSITÉ CÔTE D'AZUR

A model to reduce complexity and maintain coherence between Access Control and Transmission Control policies

Yoann Bertrand [1], Mireille Blay-Fornarino [2], Karima Boudaoud [3], Michel Riveill [4]

EQUIPE SPARKS
Mai-2016 - 41 pages

Abstract : Résumé du rapport en anglais

In order to protect resources from unauthorized access and data leakage in companies, security experts and administrators can use mechanisms such as Access Control (AC) and Transmission Control (TC). Both AC and TC are based on policies that are defined, modified and revoked by these experts. However, policy management can be a time-consuming and tiresome task, especially when both mechanisms are used on large sets of users and resources. Moreover, contradictions between AC and TC policies can appear, for instance when a legitimate user is allowed to send a resource to someone who cannot access it. Such contradictions can lead to data leakage.
In this article, we first aim at studying experts feedback concerning policy definition and usage by reporting the results of a survey we have conducted among IT professionals. Based on the results of this survey, we then present a generic model that generates TC policies based on existing AC policies. This model serves several purposes. First, it takes into account the main AC models that are used in companies (i.e. *genericity problem*). Secondly, it tackles the problem of incoherences between AC and TC policies (i.e. *coherence problem*). Thirdly, it can reduce the total number of resources and subjects managed by the security policies (i.e. *complexity problem*). Finally, it takes into account the updates frequency of companies policies (*i.e. rapidity problem*).

Key-words Data security , Security policies, Access Control, Transmission Control, Data leakage, Security management

1. Laboratoire I3S (CNRS/UNS) - bertrand@i3s.unice.fr
2. Laboratoire I3S (CNRS/UNS) - blay@i3s.unice.fr
3. Laboratoire I3S (CNRS/UNS) - karima@polytech.unice.fr
4. Laboratoire I3S (CNRS/UNS) - riveill@i3s.unice.fr

A model to reduce complexity and maintain coherence between Access Control and Transmission Control policies

Résumé : Afin de protéger les fuites de données et les accès non-autorisés, les experts sécurité et les administrateurs peuvent mettre en place des mécanismes de contrôle d'Accès (Access Control ou AC) et de contrôle de Transmission (Transmission Control ou TC) au sein des entreprises. Ces deux mécanismes sont basés le plus souvent sur des politiques qui sont créées, gérées et supprimées par ces experts. Néanmoins, la gestion de ces politiques peut prendre du temps et devenir complexe, surtout quand les deux mécanismes sont utilisés dans de grosses structures (problème de *complexité*). De plus, des contradictions entre les politiques d'AC et de TC peuvent apparaitre et entrainer des fuites de données (problème de *cohérence*). Dans un premier temps, ce rapport présente les résultats d'une étude qui a été menée sur les experts sécurité et les administrateurs. Cette étude donne des informations sur la volumétrie, les types de modèles d'AC les plus utilisés et le ressenti des experts vis à vis de ces mécanismes de sécurité. Dans un second temps, nous présentons un modèle qui permet de générer des politiques de TC à partir de politiques d'AC existantes. Ce modèle permet ainsi de répondre aux problèmes de *cohérence* et de *complexité* tout en prenant en compte la vitesse de mise à jour des politiques existantes (probléme de *rapidité*).

Mots-clefs : Sécurité des données, Fuite de données, politiques de sécurité, Contrôle d'Accès, Contrôle de Transmission, Gestion de la sécurité

1 Introduction

Adding coherent and flawless security to an IT infrastructure is not an easy task. To tackle this problem, one can start by controlling access to certain resources by using Access Control (AC) mechanisms. These well known mechanisms offer good security but they do not cover what can happen to resources (for instance documents) once they are accessed. Indeed, an authorized user can access (i.e. read) a document and then retransmit it to an unauthorized user or third party. If this third party does not have access to this document, this retransmission can be viewed as a data leakage, which can have a very bad impact on company business, wealth, image and employees.
To overcome this issue, Transmission Control (TC) mechanisms, such as Data Leak/Loss Prevention (DLP), can be added to existing AC. Both AC and TC are based on policies that define "*Who can access what ?*" (in the case of the AC) and "*Who can send what to whom ?*" (in the case of the TC mechanisms).
However, these policies can be defined separately, at different time and by different persons, leading *de facto* to incoherences. Indeed, imagine an AC policy containing a rule mentioning that *"user Tom can access (i.e. read) the resource docA.pdf"* and a TC policy specifying a rule saying that *"Tom can send all pdf files to other employees"*. If Tom retrieves docA.pdf from a secure storage and wants to transmit it to Kate (who does not have access to documentA.pdf in the AC policy), several remarks can be made.
First of all, the fact that Tom can read docA.pdf does not violate the AC policy. Secondly, the fact that Tom sends docA.pdf to Kate does not violate the TC policy. Nevertheless, the transmission will cause a violation of the AC policy because Kate will have access to this document in the end.
This simple, but yet explicit example, shows that even if both AC and TC policies seam correct, both can be in contradiction with each other, leading to potential data leakage.
Moreover, contradiction between AC and TC policies can be even more complex to underline, especially in companies with dozens of employees and thousands of documents. Thus, create, manage and keep the coherence between these policies can become a tiresome task. To have insights on this tiresomeness, we have conducted a survey that aims at gathering information on security policy management inside companies (Objective O1).

Thanks to this survey, challenges have been underlined. These challenges aim at :

- addressing the *genericity problem* (i.e. take into account several models that are commonly used in companies). (Challenge **C1**)
- addressing the *coherence problem* (i.e. AC and TC that are not coherent with each other) (Challenge **C2**).
- addressing the *complexity problem* (i.e. reduce the total number of resources and subjects managed by the policies) (Challenge **C3**).
- addressing the *rapidity problem* (i.e. take into account the time-consumption criterion to match the updates frequency of existing AC policies). (Challenge **C4**).

Our second objective is to tackle these challenges (Objective O2).

The rest of the paper is organized as follows. In the next section, existing works regarding Access Control, Transmission Control and IT experts surveys are presented. Section 3 presents the survey we have conducted to tackle Objective O1. Section 4 describes the context and

the vocabulary we are using, while section 5 and 6 give insight on our contribution, which tackles Objectives O2. Section 7 details our experiments, while discussions and conclusion are presented in Section 8.

2 Related works

This section presents the main Access Control models and Data Loss/Leak Prevention (DLP) notions. Then, existing solutions that aim at linking both AC and TC paradigms in common models or frameworks are presented. Finally, the last part gives an overview of existing surveys that have been conducted on security experts and IT administrators.

2.1 Access Control Models (AC)

Access Control (AC) aims at restricting access to resources. Several contributions have been made to create efficient and fine-grained AC models and mechanisms. The following subsections present these contributions.

Discretionary Access Control (DAC) versus Mandatory Access Control (MAC)

In the 1980's, U.S. Department of Defense (DoD) has defined the Trusted Computer System Evaluation Criteria [5] (TCSEC) [1]. TCSEC is a set of security guidelines and standards defining two different models of Access Control : Discretionary Access Control (DAC) and Mandatory Access Control (MAC). In DAC models, users can set, modify or share the access control of their resources. Most modern operating systems such as Windows, GNU/Linux and Mac OS are based on DAC models.
On the contrary, MAC refers to a family of models where owners do not have to choose the rights over their resources. In this type of access control, the system assigns security labels or classifications to resources (for instance "classified", "secret" or "top secret") and allows access to subjects or applications depending on their level of clearance. Thus, MAC are often used in military infrastructures.
As one can see, what differentiates MAC and DAC is the entity who defines the access rights. In the case of MAC, system sets the rights, while in DAC, this task is left to the discretion of users.
Over the years, many other models have been implemented and have broaden the MAC/DAC taxonomy. These models are presented in the next subsections.

Access Control Lists (ACL)

Access Control Lists (ACL) [2] have been invented before TCSEC guidelines. They were first used within 1970's multi-users systems. ACL consist of a list of entries that informs about the access rights that each user has to a specific resource. Traditional representation for ACL is a two-dimensional matrix, where columns represent resources, rows represent subjects, and intersections represent the action that the corresponding subject can perform on the corresponding resource. Such matrix is called Access Matrix and was first introduced in [3].

5. TCSEC is also known as the "Orange Book", due to its colored cover.

Role Based Access Control (RBAC)

RBAC [4] is based on the notion of role. A role is a set of subjects that share common attributes (for instance, a role "security-staff" containing all the security operators of a company). In this model, being a member of one or more groups gives users an access to certain resources.

Attributes Based Access Control (ABAC)

NIST defines Attributes Based Access Control (ABAC) as "*An access control method where subjects requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions*" [5]. Attributes can encompass various criterions about a subject (position, zip code, etc.) or an object (resource security level, type, etc.). For those reasons, ABAC is often seen as an extension of RBAC.

Policy Based Access Control (PBAC)

Policy Based Access Control (PBAC) [6] allows access rules to be defined and updated in a policy-oriented way. Policies (i.e. sets of rules) can be combined to determine if an access is authorized or not. Policies can target subjects, objects or environment. PBAC can be compared to a standardized version of ABAC and thus, it is adapted to governance oriented structures.

Capability-Based security

Capabilities have been first introduced in [7] and defined as "*a token, ticket, or key that gives the possessor permission to access an entity or object in a computer system.* Since then, capabilities have been used in physical systems such as the Plessey System 250 (first capability hardware and software system)[8].
Due to its nature, capability-based security can offer a solution for the problem of data retransmission. However, capability based systems had the reputation of being slow and complex and have been replaced over the years by ACL or RBAC in commonly-used systems. However, several works have been proposed to use capabilities in various systems and modern contexts, including distributed systems [10], Web [11] and software engineering [12]. Moreover, interesting works have tried to popularize such mechanism by destroying the myth built around capabilities for the last decades [13].

Other models for Multi-Level Security (MLS)

Over the years, many other AC models have been studied to provide Multi-Level Security (MLS), which is based on MAC classification (i.e. where resources are tagged with labels such as "confidential", "secret", etc.). The main MLS models tackle very specific problems such as confidentiality (Bell-La Padula) [14] or integrity (Biba [15], Clark-Wilson [17]).
Bell-La Padula (BLP) can tackle the problem of data retransmission, however, it is rather complex to set and use and label oriented. Moreover, it has been abandoned for years since Multics Operating System [16]. However, modern systems (in particular the one using SELinux) integrate such model and have been able to make professionals (re)discover BLP principles.

Traditional AC models offer mechanisms that can restrict access to resources. Nevertheless, they do not provide efficient mechanisms against data leakage. Capabilities and some Multi-Level Security models offer such solutions, but they have been abandoned over the years within companies for technical and complexity reasons. To cover the problem of data leakage, one solution can be to use Data Loss/Leak Prevention (DLP). Such mechanisms are presented in the next subsection.

2.2 Data Loss/Leak Prevention (DLP)

The following subsections present the main notions of Data Loss/Leak Prevention [6].

Definition and classification

In [18], a DLP is described as a "*system that monitors and enforces policies on fingerprinted data that are at rest (i.e. in storage), in-motion (i.e. across a network) or in-use (i.e. during an operation) on public or private computer/network*".

Policy definition

DLPs are also based on policies. These policies can help security experts and administrators preventing data leakage by defining rules such as *"send an email when user U1 sends document X to user U2"*. Industrial DLPs, such as the one provided by Symantec [7] or RSA [8] offer graphical user interfaces to generate and manage these policies.

Over the last few years, academic works have focused on improving detection methods by using machine learning [19] [20]. Moreover, other works have been proposed, to protect privacy [21] and emails leakage [22]. Closer to industrial preoccupation, [23] have proposed a framework that protects data shared between collaborative organizations. Finally, Data-driven mechanisms have been used to propose DLP-like solution [24] [25].
Nevertheless, such solutions do not provide efficient mechanisms that prevent contradiction between TC and existing AC policies. To overcome this issue, one solution can be to combine both Access Control and Transmission Control in an unified paradigm and define both aspects at the same time. Such solutions are presented in the next subsections.

2.3 Unifying AC and TC

Several works have been proposed to unify AC and TC in common formalisms or frameworks. By doing so, a security expert can define at the same time both AC and TC policies, reducing the risk of contradiction. This subsection presents the main works in the domain.

Usage Control (UCON)

[26] has proposed Usage Control (UCON) mechanisms by adding the notion of ongoing usage to AC. Thus, UCON can be seen as AC that answers the question " *Who can send what to*

6. DLPs have been described in various terms, including Information Leak Detection and Prevention (ILDP), Information Leak Prevention (ILP) or Content Monitoring and Filtering (CMF). Nevertheless, DLP is the most commonly used name.

7. `https://www.symantec.com/data-leak-prevention/`

8. `http://www.emc.com/security/rsa-data-loss-prevention.htm?fromGlobalSelector`

whom ?". Based on the notions of Authorizations, Obligations and Conditions, UCON offers a unified framework that covers traditional AC models and enhance them to tackle prerequisites within distributed and network-connected environments. UCON has been followed by many works that cover policy definition[27] or existing enterprise mechanism enforcement[28].

Organization Based Access Control (OrBAC)

Organization Based Access Control (OrBAC) covers a lot of issues such as conflict detection [29] or interoperability and deployment in companies Workflows [30]. In [31], OrBAC has been enhanced to tackle information flow control problem. To do so, authors have proposed a formalized Domain Type Enforcement (DTE) that covers confinement issues. Results show that OrBAC and DTE can be efficiently used to formalize an integrate security model that takes into account both Access Control and Flow Control.

eXtensible Access Control Markup Language - Data Loss Prevention (XACML-DLP)

eXtensible Access Control Markup Language (XACML) is a XML standard that defines a declarative AC policy. In October 2014, a new version of XACML has been proposed. This version, named XACML-DLP [9], embeds both Access Control and Transmission Control in a same formalism.

Linking AC and TC in a common formalism allows security experts to define at the same time both paradigms, reducing the risk of contradiction between them. Nevertheless, these solutions :

- Do not cover the *genericity problem* (the existing AC policies need to be redefined).
- Do not cover the *coherence problem* (by defining both policies at the same time, an expert can only hope to have coherence between AC and TC, without being sure of it).
- Do not cover the *complexity problem* (by identifying similar resources and subjects in order to create clusters and thus, reduce the total number of managed entities).

2.4 Surveys on Security experts

Several surveys have been conducted with security experts. For instance, feedback about their feelings concerning the practice of Bring Your Own Device (BYOD) [10] [32]. Moreover, other works such as [33] have targeted the usage of network AC technologies and best practices. In [34], security experts have been solicited to have insights on end-users security behavior. To the best of our knowledge, no surveys have been conducted on security experts nor administrators in order to have insights on policy definition hardness and feelings toward these mechanisms.

In the next section, we present the survey we have conducted in order to fill this gap.

9. `http://docs.oasis-open.org/xacml/xacml-3.0-dlp-nac/v1.0/csprd01/xacml-3.0-dlp-nac-v1.0-csprd01.html`

10. BYOD refers to the policy of allowing employees of an company to use they own Smartphones or computers for work purpose.

3 Survey on Access Control and Transmission Control in companies - Objective O1

In this section, we present the survey we have conducted in order to fill Objective O1. As stated previously, this objective aims at gathering information on security experts and administrators feelings toward Access Control (AC) and Transmission Control (TC) mechanisms within companies.

3.1 General purpose

The main objective of the survey is to have insights on Access Control and Transmission Control policies within companies. More specifically, it aims at answering several questions regarding the general usage of AC and TC mechanisms, the size of the AC policies (i.e. the amount of targeted subjects and resources) and models that are the most commonly used within companies (ACL, RBAC, etc.). Moreover, with this survey, we want to quantity the difficulty of having to define such policies. Finally, we would like to know if the people we have surveyed are interested in features that, we think, could be beneficial for administrators and security experts.

3.2 General information

The survey is a 20 questions Google Form, available on the Internet [11]. We have implemented the survey in two different languages (english and french). Both were proposed to professionals thanks to social medias (LinkedIn, Twitter, security forums and personal contacts lists).
Concerning the survey itself, we have mostly used multiple choices questions and Likert scale (i.e. notation from 0 to 5). After two months, we have gathered 30 participants' answers. These results are presented in the next subsections.

3.3 Participants' position

The first question the survey tries to answer is about the participants position. Results in **FIGURE 1** show that most participants are software administrators and engineers (77%), followed by security experts (20%). Finally, network administrators are anecdotal (3%).

3.4 Existing Access Control information

Questions about the usage of AC and the general size of AC policies (i.e. number of subjects and resources managed within these policies) have been asked. More specifically, we have asked professionals the type of model they were using.
Results in **FIGURE 2.A** show that all participants use AC policies. More specifically, it seems that traditional ACL (54%) and RBAC (13%) are the most commonly used models, while ABAC is less used (with or without other models). Finally, we underline that no other models have been suggested by participants, despite an open text box, allowing them to enter other type of AC (i.e. MAC, DAC, etc.) (**FIGURE 2.B**).

11. `https://docs.google.com/forms/d/1ZSwt-r37X5ehhOT3IFEJWC7IcWA7dcckHg1LTJEPWHs/viewform?usp=send_form`

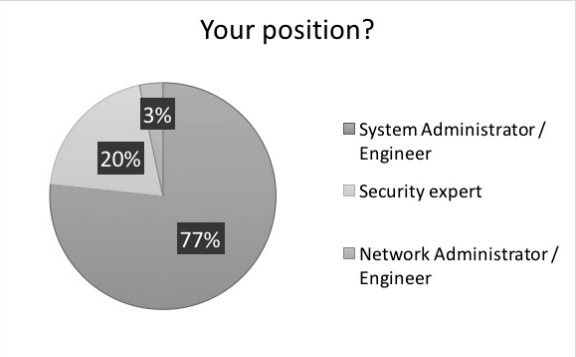
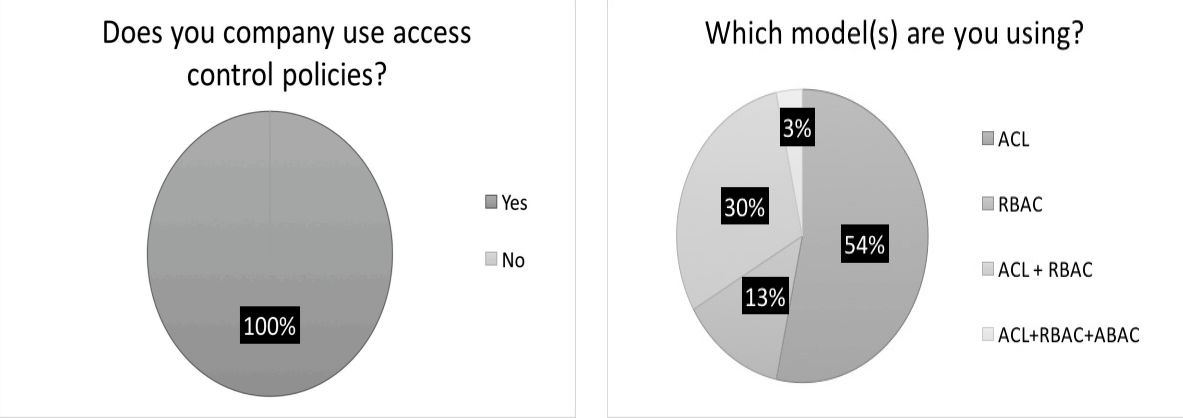


FIGURE 1 – Distribution of the 30 participants in terms of positions.



(A) (B)

FIGURE 2 – Usage of AC policies (A) and most commonly used models (B).

These results reveal a first challenge (Challenge **C1**), which is to have a solution that must take into account several models that are commonly used in companies (**Challenge C1**).

3.5 Usage and coherence of Transmission Control policies

Questions about the usage of Transmission Control mechanism have been asked. Results in **FIGURE 3.A** show that 53% of participants use Transmission Control policies. Moreover, these TC policies are often defined based on AC policies in order to have coherence between the two paradigms. Indeed, **FIGURE 3.B** shows that 69% of participants have answered that they define TC based on AC. **FIGURE 3.C** shows that people who have declared that AC and TC are kept coherent with each other have stated that having to keep this coherence is an annoying and hard task.

Moreover, we have asked participants if they were interested in a mechanism that defines TC policies based on existing AC. **FIGURE 4.A** shows that most of the people are interested by this mechanism. Finally, question has been asked regarding coherence between AC and TC (i.e. when one is modified, the other automatically adapted). Results in **FIGURE 4.B**

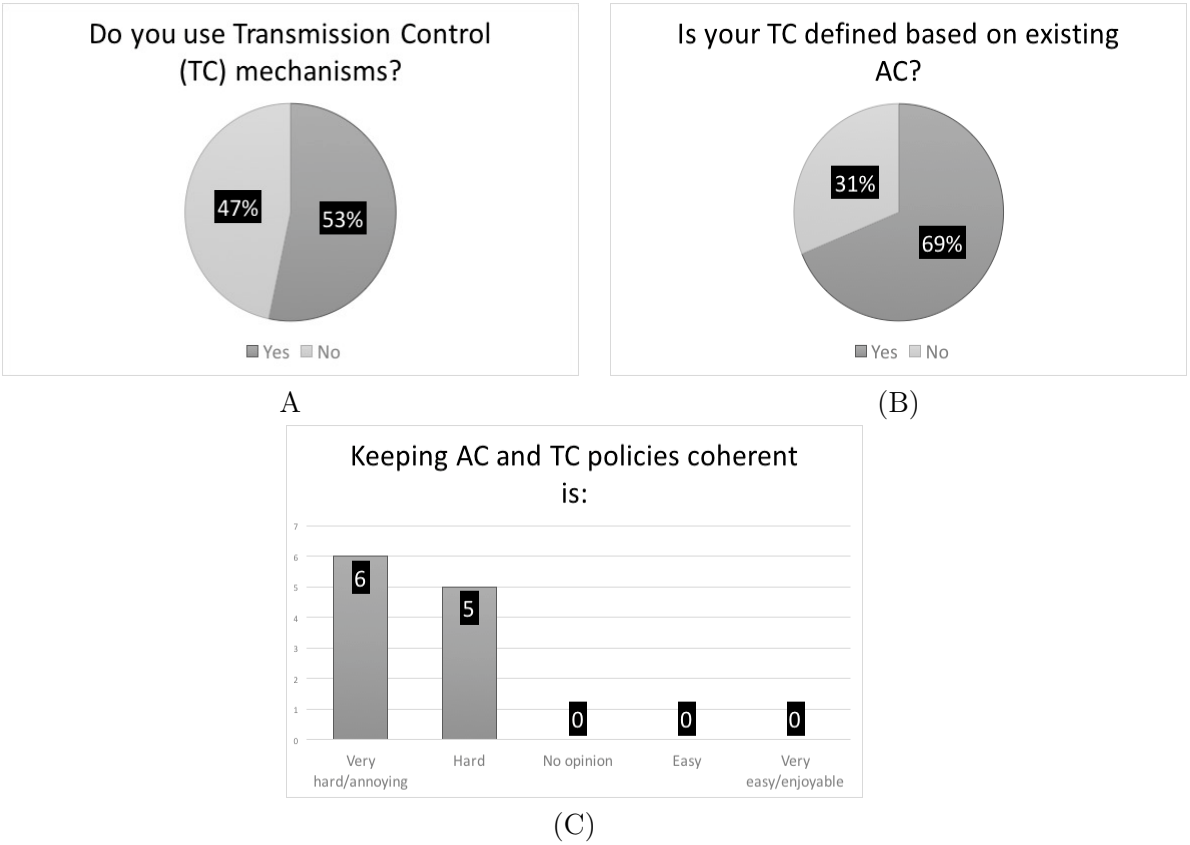


FIGURE 3 – Results concerning the usage of Transmission Control policies and link between AC and TC.

show that having a mechanism to keep coherence between AC and TC is interesting for most participants.

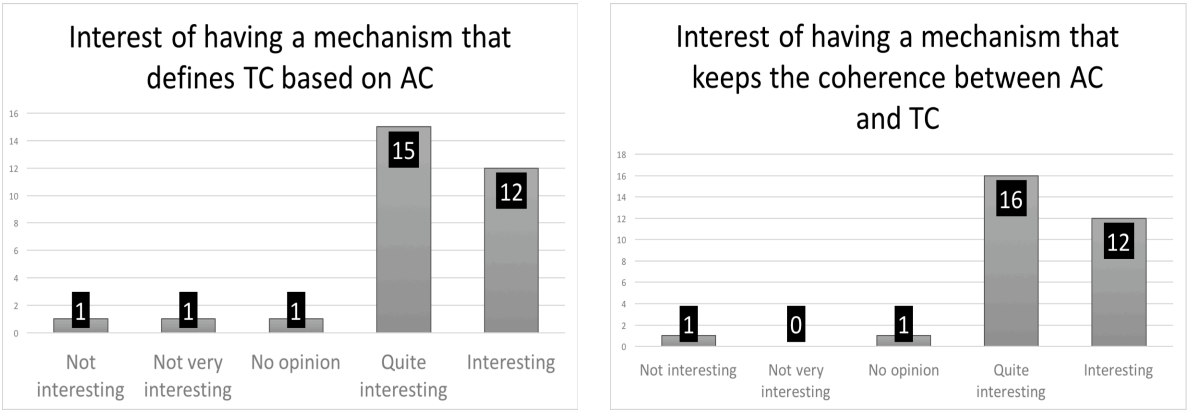
These results have underlined a second challenge (**Challenge C2**). This challenge aims at providing a mechanism that generates TC rules based on existing AC policies to reduce the tiresomeness of having to define one based on the other. Moreover, this mechanism needs to maintain coherence between AC and TC.

3.6 Volumetry of AC policies

From the volumetric point of view, results in **FIGURE 5.A** show that most AC policies target middle-sized companies or infrastructures (80% of infrastructures embed less than 250 subjects). Nevertheless, bigger infrastructures are also represented (20%).

Answers about the number of resources are more heterogenous. Indeed, **FIGURE 5.B** shows that very few AC policies manage either more than ten thousands resources (3%) or millions of them (3%). On the contrary, participants have replied that most of their AC policies manage up to few thousands resources (67%).

We have asked if having to define AC policies is a tiresome task. As it can be seen in **FIGURE 6.A**, results show that most participants have declared that it is quite the case. Except for the participants who have no particular opinion, most of the others have agreed that too



(A) (B)

FIGURE 4 – Interest of having 2 mechanisms. The first one generates TC based on AC. The second one keeps both AC and TC policies coherent with each other.

Challenges	Description
C1	To take into account several models that are commonly used in companies
C2	To take care of the *coherence problem* (i.e. AC and TC policies needs to be coherent with each other)
C3	To take care of the *complexity problem* (i.e. reduce the number of managed entities)
C4	To take into account the time-consumption criterion (in order to fit the frequency updates of existing AC policies)

TABLE 1 – Summary of the main challenges raised by the survey.

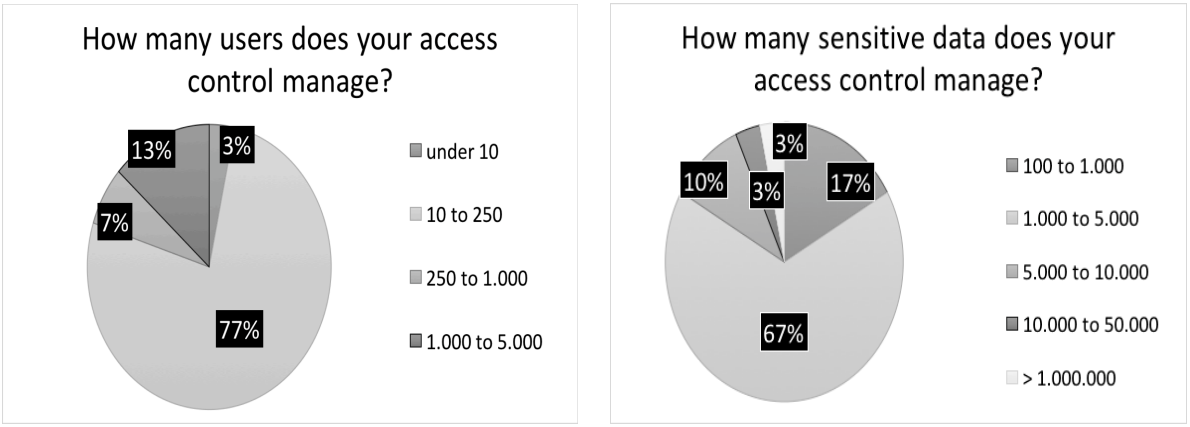
many entities are managed by their AC mechanisms (see **FIGURE 6.B**). We think that this profusion of entities might explain why many participants consider policy definition and management to be tiresome.

Finally, we have asked participants if they were interested by a feature that could reduce the total number of managed entities (i.e. subjects and resources). Results in **FIGURE 7** show that for most of them, this feature seams quite interesting.

Thanks to these results, we have identified another challenge (**Challenge C3**). This challenge aims at providing a solution that reduces the tiresomeness of having to manage too many entities.

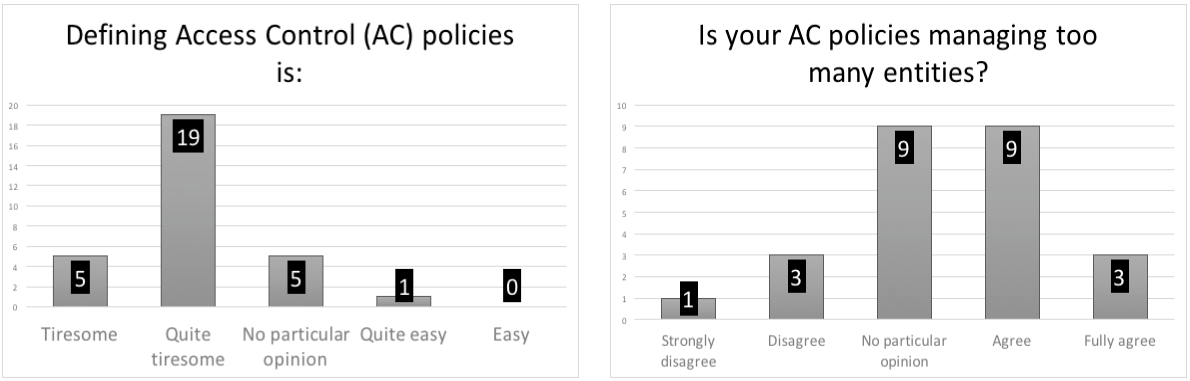
3.7 Updates frequency

The last question that we have asked concerns the frequency of AC updates. Results show heterogenous responses (see **FIGURE 8**). Nevertheless, we can conclude that updating AC is an operation that needs to be performed at least several times a day (87%).
These results reveal a last challenge (**Challenge C4**). This challenge is to provide a solution that is time-efficient and reactive regarding participants update frequency.



(A) (B)

FIGURE 5 – Volumetry of participants' AC policies



(A) (B)

FIGURE 6 – Feelings towards the AC policies regarding tiresomeness of policy definition and volumetry.

In this section, we have discussed the survey that we have conducted. Thanks to this survey, we have obtained insights about the people who have in charge the definition and maintenance of security policies within companies. The conducted survey has shown that these tasks are often done by system administrators and engineers. Moreover, the survey has revealed that traditional AC models such as Access Control List (ACL) and Role-Based Access Control (RBAC) are mostly used. For most of them, these models encompass several dozens of subjects (from 10 to 250 for 77% of participants) and a few thousands resources (1000 to 5000 for 67% of participants). Finally, answers about update frequency have shown that AC policies updates are quite frequent within companies (more than several times per hour for 2/3 of participants). In addition, this survey has revealed that Transmission Control policies are quite used within companies (53%) and that most of the participants who use both paradigms want to keep a coherence between AC and TC policies. Unfortunately, this coherence has a price, and results show that all participants think that having to keep coherence between AC and TC is quite hard and annoying.

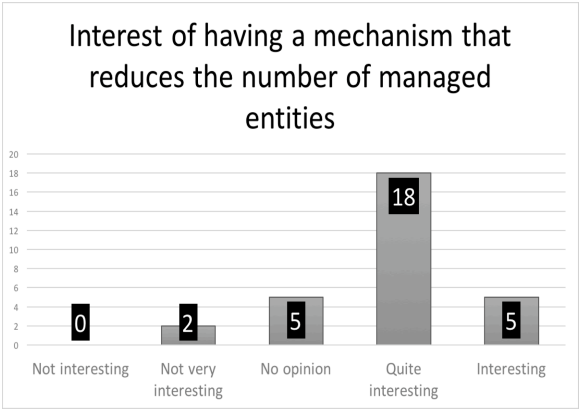


FIGURE 7 – Interest of having a mechanism that reduces the total number of managed entities.

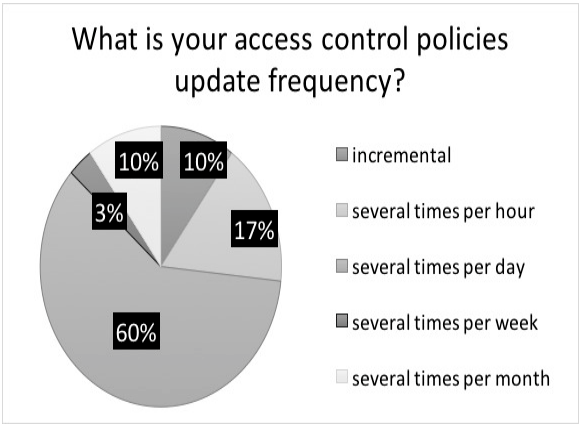


FIGURE 8 – Update frequency of the AC policies.

Finally, we have identified 4 Challenges, summarized in **Table I.**. To take up these Challenges, we have defined a model. This model is described in the next sections.

4 Provide a model to tackle the *genericity problem* - Challenge C1

In this section are introduced the context and vocabulary of our work. First subsection presents the scope of our work while the next subsections introduce our generic model. This generic model aims at taking into account many existing AC models in order to tackle challenge **C1**.

4.1 Scope of our work

Previous works [35] have introduced a first version of our model. In this paper, we have broaden this model to embed sets of actions and represent sets (i.e. clusters) of subjects and

resources. Moreover, we have been inspired by capability-based systems and have decided to add a similar concept to our model.

4.2 Generic Access Control Model

To take into account the main models used by the participants of our survey, we have been inspired by [36], and more specifically by [37] and [38] to represent generic AC as a set of rules (1). A rule is composed of three fundamental things : a Subject, one or several Actions and a Resource (2).

$$GenericAC =< \sigma_1, \sigma_2, ..., \sigma_n >, \forall \sigma \in Rules \tag{1}$$

$$\sigma = \{s, < a_1, a_2, ..., a_n >, r\}, s \in Subject, a_i \in Action, r \in Resource \tag{2}$$

Subjects, Actions and Resources are subsets of Entity (3). In our model, an entity is represented by a unique identifier (for instance a name) and a set of attributes (4). An attribute represents for instance a position (ex : role="CEO") or a security level (ex : securityLevel = "confidential"). An identifier must be unique (5).

$$\{Subject, Action, Resource\} \subset Entity \tag{3}$$

$$entity = \{identifier, < att_1, att_2, ..., att_n >\} att_i \in Attribute \tag{4}$$

$$\forall e_i, e_j \in Entity, e_i(identifier) \neq e_j(identifier) \tag{5}$$

Our model represents parameters as a pair of key/value (6). For a particular entity (ex : subject "Leslie"), two parameters cannot have the same key (7).

$$att =< key, value > \tag{6}$$

$$\forall (att_i, att_j) \in Attribute^2, att_i(key) \neq att_j(key) \tag{7}$$

Moreover, duplicates (i.e. two identical rules) cannot be contained in the same generic AC (8).

$$\begin{aligned} &\forall \sigma_1 \in GenericAC, \\ &\nexists \sigma_2 \in GenericAC / s_1 = s_2 \wedge r_1 = r_2, \\ &s_1, s_2 \in Subject, r_1, r_2 \in Resource \end{aligned} \tag{8}$$

ACL representation

Like any ACL, our generic AC can be represented as a two-dimensional matrix where rows represent the subjects and columns represent resources. The row/column intersections represent the action that can be performed by a particular subject on a particular resource. **FIGURE 9** represents such matrix.

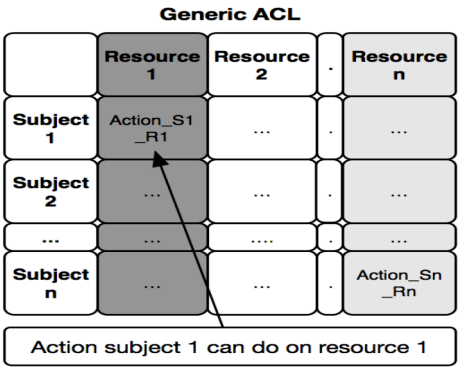


FIGURE 9 – Graphical representation of a generic ACL.

Model genericity

With our formalism, traditional AC models used by the participant of our survey (i.e. ACL, RBAC and ABAC models) can be represented. Indeed, all of these models can be represented as a set of rules. To give an example, let us take the following ABAC rule : "*every security expert can access and modify documents marked as confidential* ". This rule is equivalent to an enumeration of rules (i.e. Cartesian product) among set "security experts" and the set containing all documents marked as confidential (see **FIGURE 10**)

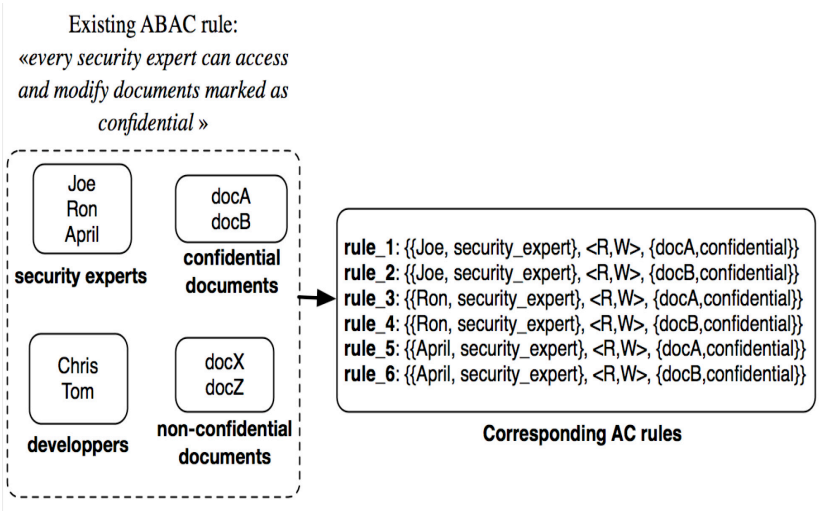


FIGURE 10 – Link between the ABAC rule "every security expert can access and modify documents marked as confidential" and the generic AC rules in our formalism.

In this section, we have presented the vocabulary that we used for our generic AC model. The goal of this generic AC model is to represent usual AC models to take up challenge **C1**.

5 Provide a mechanism to address the *coherence problem* - Challenge C2

In this section, we present our model. The main objective of this model is to tackle the problem of coherence. To do so, we propose a mechanism that can generate TC policies based on existing AC rules. First subsections present our Transmission Control paradigm and representation. Then, we present the generation mechanism that transform Access Control policies into Transmission Control policies.

5.1 Transmission Control List

Transmission Control models aim at answering the question : *"who can send what to whom ?"*. To do so, we have defined a Transmission Control List (TCL), described as a set of transmissions regarding one or more resources (9). Thus, a TCL is always related to at least one resource.
In terms of access and retransmission, two resources are equivalent if and only if they share the same set of transmissions (10). A transmission embeds the following elements : a source subject (i.e. the sender), a destination subject (i.e. the receiver), the actions of the sender and the receiver and a transmission type (11). Details about these elements are given in the next subsections.

$$\begin{aligned} &\forall tcl \in TCL, \\ tcl = \{ &< r_1, r_2, ..., r_n >, < \tau_1, ..., \tau_n > \} \\ &r_i \in Resource, \tau_i \in Transmission \end{aligned} \tag{9}$$

$$\begin{aligned} &\forall r_1, r_2 \in tcl \\ r_1(< \tau_1, ..., \tau_n >) = r_2(< \tau_1, ..., \tau_n >) &r_1, r_2 \in Resource \end{aligned} \tag{10}$$

$$\begin{aligned} \tau = < &sender, receiver, senderActions, \\ &receiverActions, transType >, \\ &sender, receiver \in Subject, \\ &senderActions, receiverActions \in Actions, \\ &transType \in TransmissionType \end{aligned} \tag{11}$$

Transmission Type

A transmission between a sender and a receiver is referred as a transmission type. A transmission type represents if a transmission is authorized without any medium transformation (TRANSMISSION_AUTH) or if the transmission is denied (TRANSMISSION_DEN).
A medium transformation means adding security properties, like confidentiality (TRANSMISSION_CONF) or Integrity (TRANSMISSION_INTEG) to the transmission. We underline that other transmission types can be defined in the model. Transmission types have two properties : non-reflexivity and completeness. Non-reflexivity means that a subject cannot send a

resource to herself/himself (this is arbitrary and can be changed). In this case, a special transmission type (represented as '-') is used. Completeness property means that a transmission must contain one and only one transmission type (12).

$$\begin{aligned} &\forall \tau \in Transmission \\ &\forall transType \in TransmissionType \\ &\exists \tau(transType)/Card(\tau(transType)) = 1 \end{aligned} \tag{12}$$

TCL Representation

A two-dimensional matrix can be used to represent a TCL (**FIGURE 11**). Rows of the matrix represent senders while columns represent receivers. Intersections represent the transmission type (for instance, TRANSMISSION_AUTH) between the sender and the receiver. We underline that actions of senders and receivers are also conserved in the TCL, due to the formalism defined in (11).This matrix can also be mapped to graph, where vertices represent transmission while each subject is represented as a node.

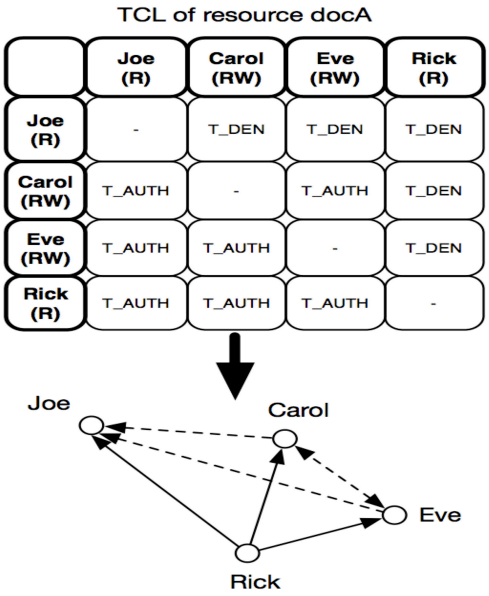


FIGURE 11 – Graphical representation of a TCL and its graph. Inside the matrix, rows represent senders and columns represent receivers.

5.2 Compute additional information

Based on the TCL formalism, additional information can be computed. This additional information is presented in the next subsections.

Node Types

A node type represents the type of node a subject can be. Node type comes in 9 flavors, depicted in **FIGURE 12**. No connection between two nodes means that the transmission is

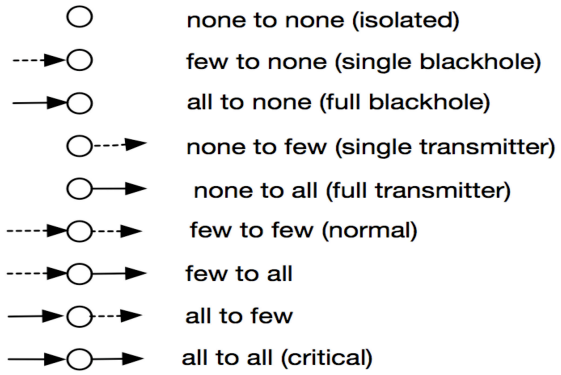


FIGURE 12 – Graphical representation of the different node types.

denied as both sender and receiver. Concerning our graphical formalism, a straight line is used when a subject can send/receive a resource from/to anyone else $(n-1)$ (when n is the total number of subjects allowed to access a resource). On the contrary, a dashed line represents a node type that can send (or receive) a resource at least from one subject, but not from all of them (i.e. between 1 and $n-2$).
As an example, graph in **FIGURE 11** shows that Joe can receive the resource from everyone while he cannot send it to anyone. Thus, Joe has a "*full blackhole*" node type for this particular TCL.

Node type allows IT professionals such as security experts or administrators to prevent resource propagation inside an infrastructure. Indeed, by giving capabilities to subjects, one can prevent a subject to retransmit a document by giving her/him a node type that cannot send the resource (for instance a single blackhole or a full blackhole).

Capabilities

In the ACL, subjects can perform one or more actions over resources. In the TCL representation, we introduce the concept of capability. As stated in related works (2.1.6), capabilities are token possessed by users that allow them to perform actions. In our model, a capability is the combinaison of an ID, an action (ex : Read) a node type (ex : Full blackhole) and a list of resources the capability is applied to (13) :

$$\begin{aligned} Capability = identifier, action, nodeType, \\ < res_1, res_2, ..., res_n > \\ action \in Actions, nodeType \in NodeTypes \\ \forall res_i \in Resources \end{aligned} \tag{13}$$

Thus, a capability represents what a subject can do in terms of Access Control (action) and in terms of Transmission Control (nodeType). A subject can have several capabilities for a particular resource, but each of her/his capabilities will have the same node type for this resource. For instance, subject Carol for the matrix in **FIGURE 11** will have the following capabilities :

Carol = {<001, Read, normal, docA> , <002, Write, normal, docA>}

If Carol can also access resource docB with Read permission and she can receive it from everyone else, but cannot send it to anyone, she will have the following capabilities :

Carol = {<001, Read, normal, docA> , <002, Write, normal, docA> <003, Read, full_blackhole, docB> }

Thus, our model represent capabilities as the type of access control and transmission control subjects can perform over resources.

5.3 Coherence definition

Now that ACL and TCL have been defined, coherence can be introduced. We define coherence based on 2 principles :

— **P1** : For every subject in the ACL, a subject must appear at least in one TCL (14).
— **P2** : If a subject can do certain actions on a resource in the ACL (such as read and write), she/he will have the exact same actions on this resource's TCL (15).

$$\begin{aligned} P1 : \forall s \in acl, \exists s \in tcl \\ s \in Subject \\ acl \in ACL, tcl \in TCL \end{aligned} \tag{14}$$

$$\begin{aligned} P2 : \forall (s,a) \in acl, s \in Subject, a \in Action \\ \exists (s,a) \in tcl \\ acl \in ACL, tcl \in TCL \end{aligned} \tag{15}$$

Thus, Coherence C between AC and TC policies is true if and only if P1 and P2 are true (16).

$$C \Rightarrow P1 \wedge P2 \tag{16}$$

Now that all aspects of our model have been presented, we introduce in the next subsections the mechanism we have implemented to transform a generic ACL into TCLs.

5.4 Generation Mechanism

This section focuses on the generation mechanism aiming at transforming existing ACL into TCLs. This subsection presents the main parts of this mechanism.

Creation of the TCL structure

In order to create the general structure of TCLs, the mechanism starts by retrieving all resources described within the ACL. For each resource, the mechanism retrieves every subject that has an explicit access right to this resource (we call such subjects "*marked subjects*").

Then, the mechanism creates the general structure of the matrices (one matrix per resource) by adding for each row and column the marked subjects as sender and receiver. Then, actions are copied without any modification in order to respect the second principles of coherence defined in (14) and (15). Thanks to this mechanism, coherence is respected and challenge **C2** is covered.

Finally, we underline that the size of the TCL depends on the number of marked subjects. Indeed, a resource that can be accessed by many subjects in the ACL will generate a bigger TCL than a resource with fewer marked subjects.

Mapping Rules Concept

Once the TCL's structures have been generated, they have to be filled. To do so, we have formalized the concept of Mapping Rules (MR). A MR can be represented as a function that takes parameters of a sender, actions, receiver and resource and returns a transmission type (17).

$$\begin{aligned} f(&sender, senderAction, receiver, \\ &receiverAction, resource) \\ &\rightarrow type \\ &type \in TransmissionType \end{aligned} \tag{17}$$

For each element of the matrix (i.e. each row/column intersection), a mechanism retrieves all the parameters that concern the sender, the receiver, the action of the sender, the action of the receiver and the resource, then output a transmission type. This mechanism is graphically represented in **FIGURE 13**.
To define how the transmission type is chosen depending on the entry parameters, we have defined a specific syntax that allows security experts to express mapping rules. Details about this syntax are given below.

Mapping Rules Syntax

To define Mapping Rules, we have defined a particular syntax, called Mapping Rules Syntax (MRS). MRS is based on three different elements : targets, operators and inputs. A target can be formalized as an entity and an element (18). An entity can be a sender, a receiver, an action of the sender, an action of the receiver or a resource (19). An element can be represented as an entity identifier (ex : "Ron"), a parameter key (ex : "role") and a parameter value (ex : "manager")(20).

$$target = (entity, element) \tag{18}$$

$$\begin{aligned} entity = \{&sender, receiver, \\ &senderAction, receiverAction, resource\} \end{aligned} \tag{19}$$

$$\begin{aligned} element = \{&identifier, \\ &parameter(key), parameter(value)\} \end{aligned} \tag{20}$$

MRS uses two types of operators : arithmetic operators and logical operators (21) :

$$\begin{aligned} arithmeticOperator &= \{=, \neq, <, >, \geq, \leq\} \\ logicalOperator &= \{\vee, \wedge\} \end{aligned} \tag{21}$$

Finally, the last component of MRS is the input, which is just a String (i.e. any word in the alphabet $\mathcal{A}$) (22).

$$input \in \mathcal{A}^* \tag{22}$$

Thanks to previous definitions, security experts can define two types of rules : comparative rules and specific rules. A comparative rule is composed of a target, an arithmetic operator, another target and a resulting transmission type (23) :

$$\begin{aligned} comparativeRule = senderTarget,& \\ arithmOp, receiverTarget \rightarrow type& \\ senderTarget, receiverTarget \in Target& \\ arithmOp \in ArithmeticOperator& \\ type \in TransmissionType& \end{aligned} \tag{23}$$

Comparative rules can be used to provide predefined and generic patterns to security experts. For instance, a generic rule could be that "*you cannot send a resource to someone with a lower accreditation level than yours*". Considering that subjects have a parameter "level" describing such accreditations, the previous generic rule will be :

rule1 : (sender, level) > (receiver, level) → TRANSMISSION_DEN

This generic rule will then be applied to all row/column intersections, for all generated TCLs structures.

To provide fine-grained rules, one can use specific rules. A specific rule is defined by a target, an arithmeticOperator and an input (24) :

$$\begin{aligned} specificRule = senderTarget,& \\ arithmOp, input \rightarrow type& \end{aligned} \tag{24}$$

Specific rules are used to define specific conditions on parameter values (for instance, "action = Read"). A specific rule such as "*deny the transmission if the receiver is Ann*" will be represented as follows :

rule2 : (receiver, identifier) = "Ann" → TRANSMISSION_DEN

Moreover, conditions can be combined with logical operators to express more complex rules. For instance, the broad concept of "*if sender and receiver can both read and write the resource, they can transmit the resource to each other*" can be represented as :

rule3 :
(sender, identifier) = "Garry" ∧
(resource, identifier) = "docZ.pdf" ∧
(receiver, identifier) = "developers" → TRANSMISSION_AUTH

However, combining rules can lead to multiple transmission types results. However, we have defined in (12) that this is not possible. Thus, we have defined a conflict detection to overcome this issue. This conflict detection is presented in the next subsection.

Conflict detection

Our model allows a security expert to express various transmission rules. Nevertheless, mapping rules definition and combinaison can lead to conflicts. Indeed, imagine for instance that a security expert defines two different rules **r1** and **r2**, where **r1** defines *"When managers are sending a resource, the transmission must have confidentiality property"* and **r2** defines *"John cannot send docA.pdf"* (even if he has access to it in the ACL). Imagine now that John is a manager. The mapping mechanism will have issues deciding which transmission type to apply for every element in the row "John" for the TCL of docA.pdf. Indeed, for this resource, the system will not be able to determine if the resource can be sent (**r1**) or not (**r2**). To overcome this issue, we have defined several mechanisms.
The first one notifies the security expert of the inconsistency and ask her/him for an answer. She/he can choose the transmission type of her/his choice, or implement an *ad hoc* rule, by combining conditions for instance.

Another mechanism that we have defined is the decision strategies (DS). To use decision strategies, a security expert first needs to set levels for transmission type. For our example, we have considered that a denied transmission is more secure than the other types of transmission. Thus, we have chosen the following order :

$$\text{Level\_1 : TRANSMISSION\_AUTH} < \text{Level\_2 : TRANSMISSION\_CONF} < \text{Level\_3 : TRANSMISSION\_DEN}$$

Once levels have been defined, the security expert can use one of the following decision strategies :

— HIGHEST : apply the transmission type with the highest level
— LOWEST : apply the transmission type with the lowest level
— MOST PRESENT : apply the transmission type which is the most present in the sequence of rules
— DEFAULT : apply the default transmission type

In our example, the following rule will be applied, depending on the strategy :

Strategy	**Applied rule**
HIGHEST	r2
LOWEST	r1
MOST PRESENT	No answer, DEFAULT applied
DEFAULT	TRANSMISSION_DEN

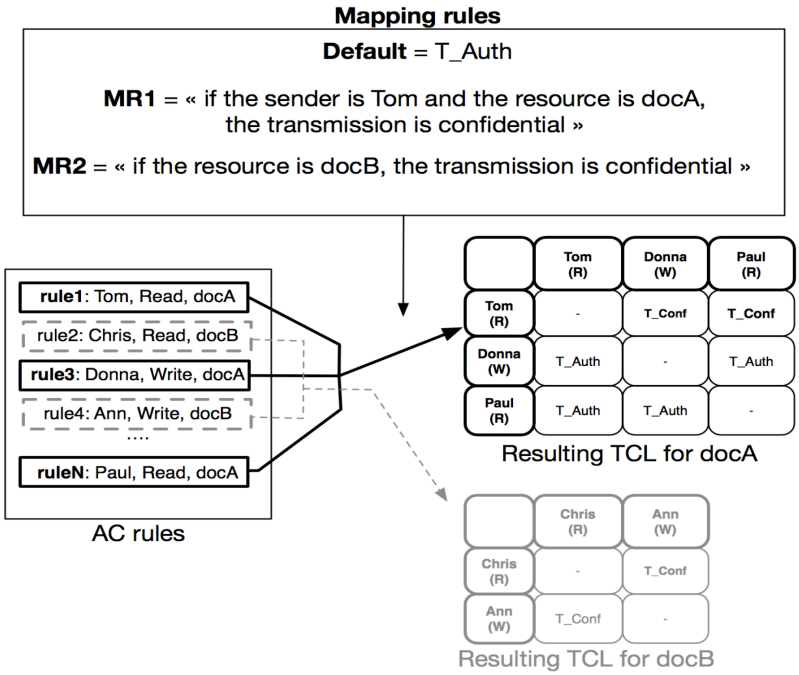


FIGURE 13 – Graphical representation of the generation mechanism.

To transform AC policies into TC policies, we have defined a specific syntax, called Mapping Rules Syntax (MRS) (see 5.4.3). Thanks to this syntax, an ACL (which can be represented as a two-dimensional matrix) can be transformed into many TCLs matrices. Each TCL represents all transmissions marked subjects can/cannot do for a specific resource.
However, we empathize that :

— The number of generated TCLs is equivalent to the total number of resources contained in the generated ACL.
— The size of capabilities sets for every single user can be consequent.

To overcome these issues and allow security experts to manage in a easier way the security of their company, we have provided inferences mechanisms that can detect similarities between subjects and resources. Thanks to these inferences, resources and subjects are clusterized, reducing *de facto* the numbers of TCLs and the complexity of capabilities sets that are managed. These inferences mechanisms are presented in the next subsections.

6 Provide a mechanism to tackle the *complexity problem* - Challenge C3

For every resource contained in the generated ACL, a corresponding TCL is generated. Unfortunately, such mechanism can generate many TCLs, depending on the total amount of resources managed by the ACL. For that purpose, we aim at reducing the number of TCLs with clustering technics, depending on the similarities of these resources. By doing so, we aim at covering the problem of complexity, and thus, tackle the challenge **C3**.

6.1 Resource similarities

First, we introduce the resource similarities mechanism. This mechanism is able to clusterize resources that have similar TCLs.

Principles

Two resources are equivalent if they can be accessed and retransmitted by the same subjects, in the exact same way. In other words, two resources are equivalent if their TCLs are identical.

Resource Cluster (RC)

A resource cluster can be described as an identifier and a set of resources that share similar TCLs (25).

$$\begin{aligned} ResourceCluster = \{identifier, < res_1, ..., res_n >, tcl\} \\ /res_1(tcl_1) = res_2(tcl_2) = ... = res_n(tcl_m) \\ \forall res_i \in Resource \\ tcl_j \in TCL \end{aligned} \tag{25}$$

A resource can be in one and only one RC (26), and the identifier must be unique (27) :

$$\begin{aligned} \forall r \in Resource \\ \exists! rc \in ResourceCluter / r \in rc \end{aligned} \tag{26}$$

$$\begin{aligned} \forall rc_i, rc_j \in ResourceCluster, \\ rc_i(identifier) \neq rc_j(identifier) \end{aligned} \tag{27}$$

To create a resource cluster, our inference mechanism takes all generated TCLs and compares them with each other. If several resources share identical TCLs, a Resource Cluster (RC) is created and these resources are put in the cluster. Then, one TCL is linked to the RC while the other similar TCLs are destroyed. In the case where a resource does not have a match, a singleton RC is created. **FIGURE 14** shows the resource similarities in a more graphical way.

6.2 Subject Similarities

Our model embeds capability concepts, allowing subject's actions to be described as a set of capabilities. Thanks to this representation, subjects can be compared and clustered thanks to our subjects similarities mechanism. This mechanism is described in the next subsections.

Principles

Two subjects are identical if and only if they can access and retransmit the exact same resources in the exact same way. In other words, two subjects are identical if they have the exact same capabilities set.

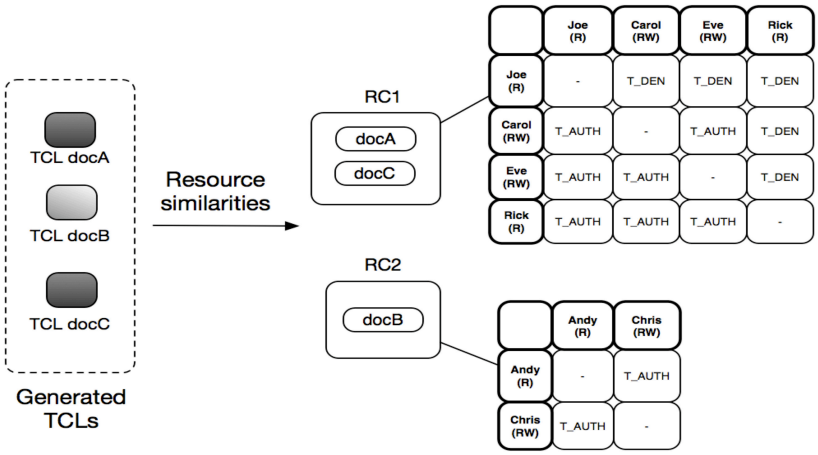


FIGURE 14 – An example of the resources similarities mechanism. Here, docA and docC have similar TCLs. Thus, a resource cluster (RC1) is created and both resources are affected to this cluster. Then one of the TCL is affected to the cluster. On the contrary, DocB does not share similarities with other resources. For those reasons, this resource is affected to a singleton RC (RC2). As it can be seen, this mechanism will have 2 TCLs (one per RC) instead of 3 (one for docA, docB and docC).

Subject Cluster (SC)

A subject cluster can be described as an identifier and a set of subjects that share the same capabilities sets (28).

$$\begin{aligned} SubjectCluster = \{identifier, < sub_1, ..., sub_n >\} \\ /sub_1(setOfCapabilities) = sub_2(setOfCapabilities) \\ = ... = sub_n(setOfCapabilities) \\ \forall sub_i \in Subjects \end{aligned} \tag{28}$$

A subject cannot be in more than one SC (29) and each identifier is unique (30). Moreover, if a subject is the only one that can access a certain set of resources in a certain way, this particular subject will be putted in a singleton cluster.

$$\begin{aligned} \forall s \in subjects \\ \exists! i \in \mathbb{N} / s \in sc_i \\ sc \in SubjectCluster \end{aligned} \tag{29}$$

$$\begin{aligned} \forall sc_i, sc_j \in SubjectCluster, \\ sc_i(identifier) \neq sc_j(identifier) \end{aligned} \tag{30}$$

To create a subject cluster, our mechanism compares every subject with each other and detect if they share the same set of capabilities. **FIGURE 15** presents the concept of subjects

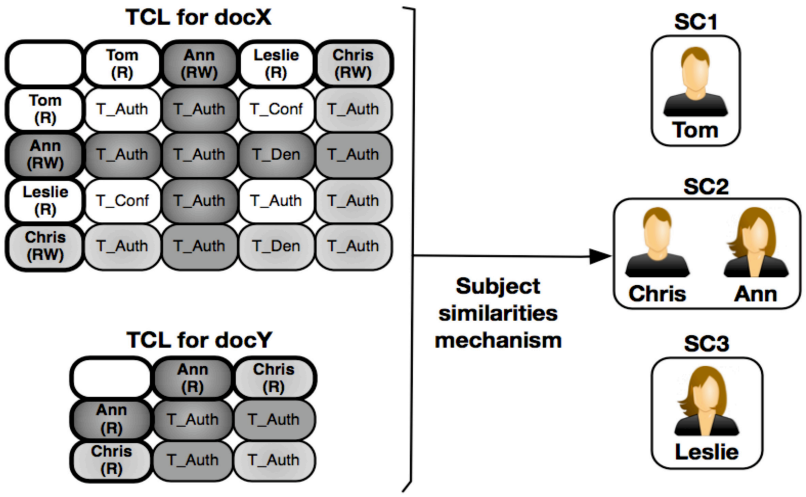


FIGURE 15 – An example of the subjects similarities mechanism. In this example, Chris and Ann can access and retransmit all resources in the exact same way (and thus, have same capabilities). Thus, the mechanism creates a subject cluster (SC2), and put both subjects in it. Other subjects, such as Leslie and Tom, does not share similarities. For those reasons, they are affected in singleton clusters (SC1 and SC3).

similarities mechanism in a graphical way.

In previous subsections, we have described our similarities mechanisms. These mechanisms aim at creating clusters of similar subjects and resources in order to reduce the total amount of TCLs and tackle the *complexity problem* (challenge **C3**).

6.3 Coherence between AC and TC policies

In previous sections, we have seen that generated TCLs are coherent with AC thanks to the principles defined in (15) and (16). However, if the ACL or one of the TCLs are modified afterwards, incoherences can appear. For instance, if a new subject is created in the TCL, she/he will not be in the ACL, violating *de facto* the first principle. Secondly, if an existing subject has new capabilities, the second principle will be violated. To cover this problem and maintain coherence (and thus Challenge **C3**) we introduce in this section a mechanism that keeps coherence of AC and TC policies even when one of them is modified.

AC operations

Our model allows a security expert or administrator to add, modify or remove an AC rule. By doing so, several modifications can occur in the TC realm.

TC operations

Because TC paradigm embeds clusters, Access Control (such as Read) and Transmission Control (such as an authorized transmission "TRANSMISSION_AUTH"), we assume that

security experts or administrators would like to manage their policies from the TC perspective rather than the AC policies. Do to so, our model allows the following operations :

— create a new subject and add her/him to a specific SC,
— create a new subject by giving a set of capabilities,
— modify a subject's capability,
— move a subject from a subject cluster to another,
— delete an existing subject,
— create a new resource and add it to a specific RC,
— create a new resource by giving a set of capabilities,
— delete an existing resource.

These operations offer security experts or administrators a way to maintain and update their Transmission Control policies in a quite easy way. But, applying modifications on subjects or resources can impact both AC and TC realms. Indeed, from the TC point of view, an action (such as creating a new subject) can modify the structure of a TCL.
From the AC point of view, adding, modifying or removing rules can also modify the existing TCL, generating incoherences between the AC and the TC realms. Thus, our coherence model automatically adapts one realm when the other is modified. The following subsections present the coherence mechanisms that we propose.

Coherence mechanisms - From AC to TC

In order to have a better understanding of the coherence mechanism, let us take two cases. First of all, imagine that a security expert decides to add an AC rule to the existing ACL. This rule is, as our AC generic model states, a combination of a subject, a list of actions, and a resource (2). The resource targeted by this new rule can either be an existing one, or a new one.
In the case where it is a new one, a new TCL will be generated. This new TCL can either be identical to a previously generated TCL, or different. In the case where the TCL is different from any other TCL previously generated, the mechanism will generate a new RC, attach the new TCL to it and put the resource in the new cluster. However, if the generated TCL is equivalent to another existing TCL, only the resource will be added in the corresponding RC and the new TCL will be destroyed.
In the case where the new rule targets an existing resource, this rule can modify the TCL by adding a new marked subject or by modifying an already marked subject capability. In these two cases, the AC rule will modify the TC realm. Indeed, the corresponding TCL will be modified (for instance, in the case of a new subject, a new row and column will be added to the TCL). These modifications will cause a cascade effect :

— **RC readjustment** : the resource targeted by the new rule will probably not be in the same RC anymore (because of TCL modifications). Readjustment needs to be done in order to re-affect the resource. This re-affectation can either be in an existing RC (if the new TCL in equivalent to an existing one), or in a new singleton RC.
— **Subject capabilities updates** : if the new rule is targeting an existing subject, there is a good chance that her/his capabilities will change, inducing an update of her/his capabilities set.
— **SC readjustment** : because of the subject capabilities update, subject membership of an existing SC can become inadequate. A readjustment is thus necessary in order to re-affect the subject in another SC. This SC can either be an existing SC (i.e. one

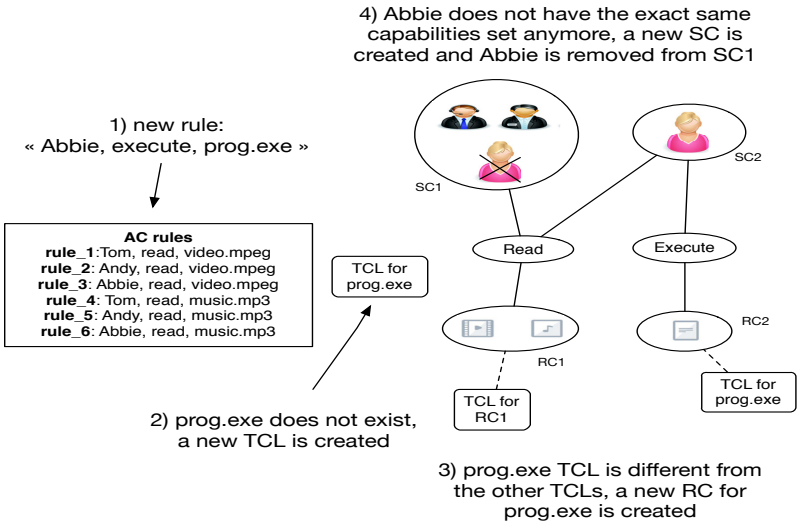


FIGURE 16 – An example of the coherence mechanism when a new AC rule is added.

SC where subjects share the exact same set of capabilities) or a new one (if no other subjects has the exact same capabilities set).

FIGURE 16 gives a graphical representation of the coherence mechanisms when an AC rule in added.

Coherence mechanisms - From TC to AC

As stated previously, we have assumed that because of the higher abstraction of our TC model, security experts and administrators can be willing to directly use TCL in order to manage security. Nevertheless, modifications in the TC realm will induce incoherences between TC and AC. Indeed, imagine that a security expert decides to use our TC model to create a new subject and put it in an existing SC. Thanks to her/his membership to this SC, the new subject will have the same set of capabilities than other members of the SC (for instance, the ability to Read-Write docA.pdf and to send it to some other marked subjects). However, there is no trace of this subject in the existing generic AC, and it will violate the coherence principles. Thus, this incoherence must be corrected. To do so, our coherence mechanism automatically generates the corresponding AC rules and add them to the existing AC policies, in order to keep coherence between the two realms. For the sake of understanding, **FIGURE 17** depicts the coherence mechanism in a more graphical way.

Previous subsections have introduced the concept of coherence mechanisms. These mechanisms allow security expert to define and manage new policies from both AC and TC realms. Unfortunately, modification of one of these two realms can generate incoherences in the other. To overcome this issue, our model automatically updates both realms to keep them coherent with each other.

In section 5 and 6, we have proposed a model that covers 3 challenges. First, we have defined a generic formalism to take into account the main AC models that are used by the

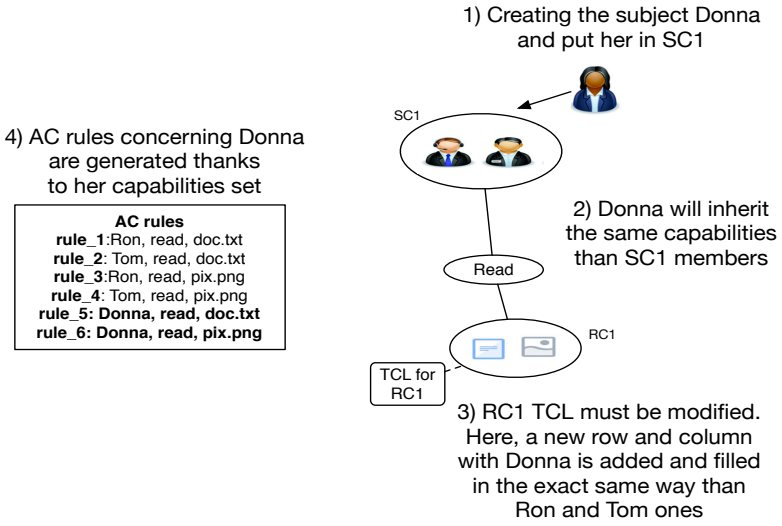


FIGURE 17 – An example of the coherence mechanism when the operation of creating a new subject in the TCL realm is processed.

participants (challenge **C1**). Secondly, our model is able to tackle the *coherence problem* by generating TC policies that are coherent with existing AC policies. This generation is done by a Mapping Rule Syntax (MRS). Moreover, a coherence mechanism has been implemented to keep this coherence when AC or TC policies are modified (Challenge **C2**).
Finally, we have tackled the *complexity problem* by offering a clustering mechanism that reduces the amount of managed entities (i.e. resources and subjects). We underline that this mechanism can be useful to ease the management for security experts and administrators (Challenge **C3**). To empirically validate the previous challenge and take on challenge **C4** (i.e. to take into account the time-consumption criterion), we have tested our model. These tests are presented in the next section.

7 Evaluation

Empirical tests have been conducted to see if the challenges we have underlined during the survey can be took up.

7.1 Implementation and tests conditions

To do our tests, we have used a MacBook Pro Retina (Intel Core i7, 2,4 GHz, 16GB RAM, 256 GB SSD hard drive) and Java 7. Java Virtual Machine has been tweaked with a heap size of 4096 bytes and artificial ACLs have been generated following the answers of the survey. Thus, we have generated ACLs that embed up to 250 subjects and 7500 resources. Concerning the actions, we have considered the Read, Write, Read-Write and Delete actions and have considered 3 parameters per subjects (name, address of the company, and job position). Information about these ACLs are given in the first 4 columns of Table 2.

In order to fill the TCL matrices with authorized or denied transmission, we did not use mapping rules, because mapping rules can greatly modify the results of resources and subjects similarities mechanisms. Indeed, if we take the mapping rule "*never allow transmission*", every single matrix will be filled with a denied transmission type (TRANSMISSION_DEN). If so, the odds to have 2 TCLs matrices that are similar will be increased, inducing biais in the results. To overcome this issue, we have decided to fill the matrices with a stochastic method. Thus, matrices are randomly filled with TRANSMISSION_AUTH, TRANSMISSION_DEN and TRANSMISSION_CONF transmission types.
Finally, all tests have been conducted 5 times. For every set of tests, minimal and maximum values have been removed and mean values have been computed based on the 3 remaining results.

Now that tests conditions have been presented, the following subsections present conducted experiments and results we have obtained.

7.2 Generation mechanism

This subsection focuses on the generation mechanism. As stated previously, the generation mechanism aims at transforming general Access Control into TCLs matrices. In order to validate Challenge **C1**, we have taken various models as examples. For instance, the following RBAC implementation rules can be translated in our formalism.

```
//RBAC Implementation
name: "prof"
    description: "profGroup"
    permission_grants:
            resource_uid: "/path/foo"
            permission_types:
               "read"

//Equivalent Generic AC rules
r1: {{Paul, prof}, <R>, {doc1,"/path/foo"}}
r2: {{Paul, prof}, <R>, {doc2,"/path/foo"}}
r3: {{Jean, prof}, <R>, {doc1,"/path/foo"}}
r4: {{Jean, prof}, <R>, {doc2,"/path/foo"}}
```

Indeed, every subject in the set prof will be able to read every resources in directory foo. In our case, Jean and Paul are professors, and directory "foo" contains 2 resources. We underline that these groups can be reconstructed by the resources similarities mechanism (if and only if similar rules are applied on both Paul and Jean).

Moreover, LDAP entries and policies can also be used. As an example, we have used an LDAP entry of our university :

```
uid=ybertran,ou=People,dc=polytech,
dc=unice,dc=fr
uid                      : ybertran
 ....
 mailacceptinggeneralid : ybertran
 mailacceptinggeneralid : Yoann.Bertrand
 cn                       : Bertrand Yoann
 role                     : etudiant
 displayName              : Yoann Bertrand
 ....
```

This information can be transformed as attributes in our model. Concerning the AC rules in LDAP, the model defines a rule as follows :

```
access to attrs=displayName
   by yoann write
```

This rule allows subject yoann to change his displayName, thus the rule can be transformed as follows :

```
r1: {{Yoann, etudiant, ybertran,...},
<W>, {displayName}}
```

Raw data presentation

In order to create all TCLs, the mechanism has to take every single resource and go through the list of subjects in order to select the marked one (i.e. subjects that have an access over this resource). Then, for every marked subject as sender and receiver, the mechanism goes through every parameter to determine if the mapping rules are applied or not (for the veracity of the results, we have maintained this part of the algorithm for the tests despite the fact that Mapping Rules were not used). We empathize that this algorithm is not the most efficient, however, we underline that computation time for empirical tests are quite sufficient. Indeed, Table 2 shows that the generation mechanism takes for most cases less than an hour to compute. Thus, this mechanism can be suitable for most participants update frequencies. For participants who have answered that their update frequencies were faster than several times an hour, we underline that the generation process only needs to be done once. Indeed, once the TCLs are generated, further modifications will be managed by the coherence mechanism, which is much more faster (see subsection 7.3). Thus, we can conclude that Challenge **C4** is validated.

ID	Rules	Subjects	Resources	Generation Time (in sec)	Resources Gain	Resources Time (in sec)	Subjects Gain	Subjects Time (in sec)
ACL_1	1000	25	300	25	45%	12	19%	0,01
ACL_2	1000	25	500	25	48%	7	24%	0,02
ACL_3	1000	100	900	5	63%	1	17%	0,3
ACL_4	1000	200	900	5	54%	2	20%	1
ACL_5	5000	20	1000	450	20%	140	28%	2
ACL_6	5000	20	2000	700	61%	80	35%	0,05
ACL_7	5000	20	3000	1800	26%	90	43%	5
ACL_8	5000	100	1000	550	53%	550	30%	65
ACL_9	5000	150	3000	1300	60%	600	33%	85
ACL_10	5000	250	1000	600	4%	750	9%	10
ACL_11	5000	250	3500	1400	55%	350	7%	12
ACL_12	7500	10	5000	1000	77%	750	30%	26
ACL_13	7500	20	1000	850	11%	1010	10%	1
ACL_14	7500	20	2000	1200	25%	280	24%	5
ACL_15	7500	20	5000	1300	70%	330	31%	2
ACL_16	7500	200	5000	2900	61%	270	28%	3
ACL_17	7500	250	3500	3600	35%	580	9%	2
ACL_18	10000	20	2000	1400	16%	330	5%	6
ACL_19	10000	50	2000	1900	21%	710	10%	2
ACL_20	10000	50	6000	4200	49%	950	15%	5
ACL_21	10000	50	7500	4500	76%	800	17%	50
ACL_22	10000	200	2000	2400	11%	3200	10%	2
ACL_23	10000	200	5000	3500	42%	2100	15%	25
ACL_24	10000	200	7500	4700	55%	3200	10%	37
ACL_25	10000	250	3500	3550	19%	4000	5%	15

TABLE 2 – Global results obtained after the tests. All results are mean values obtained after 5 experiments. Lower and higher values have been removed in order to compute the mean value with the 3 remaining results.

Interpretation

As one can notice, TCLs generation results are quite variable, even with ACLs sharing similar volumetry. These results depend on the amount of rules the TCL generation process. Indeed, the more rules an ACL will have, the more time consuming the TCLs generation will be. This can be explained by the fact that our mechanism needs to browse the ACL many times in order to retrieve subjects, resources and Access Control permissions. Thus, a bigger ACL will generate longer loop and thus, longer execution time.

7.3 Coherence mechanism

In this subsection, we describe the coherence mechanism tests. As stated previously, this mechanism aims at automatically adapt ACL or TCLs when the other is modified. In other words, instead of having to regenerate all TCLs when the ACL is modified, the coherence mechanism automatically adapts the corresponding TCLs based on the modification. Likewise, when a TCL is modified, the coherence mechanism adapts the ACL in order to keep coherence between AC and TC paradigms.
We have decided to validate this mechanism with two different approaches. First of all, we have decided to implement a graphical tool to validate that the coherence mechanism modifies the other paradigm. Secondly, we have decided to test the time-consumption of actions such as creating a new subject or adding a rule to the AC policy.

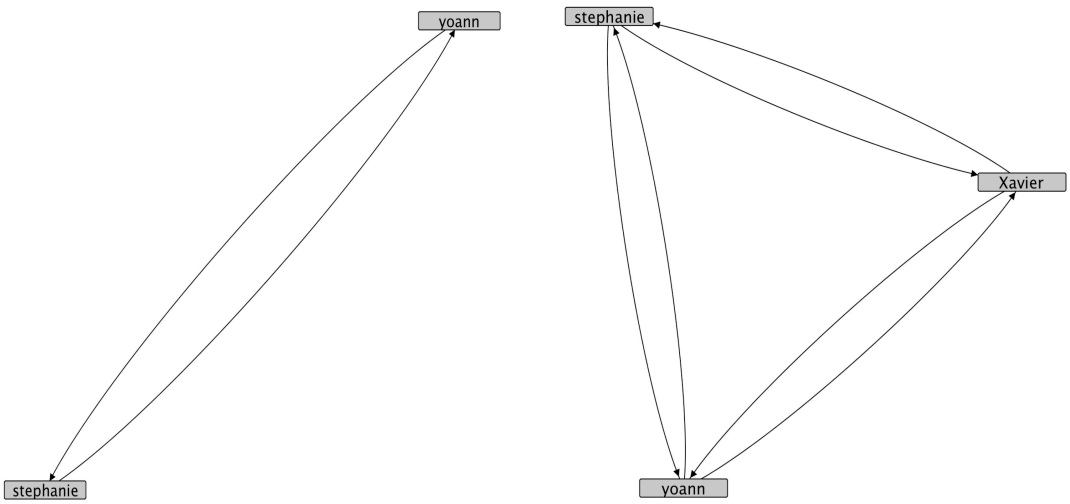
Results of these two approaches are presented in the next subsections.

```
▼<rule name="new rule">
   <subject subjectName="quentin" subjectLocation="Paris" subjectPosition="intern"/>
   <action actionName="Read" actionType="non-critical"/>
   <resource resourceLevel="confidential" resourceName="docA" resourceType="image"/>
 </rule>
```

FIGURE 18 – An example of a new AC rule added to the ACL.

Adding a new AC rule

The rule "*user Quentin can read document docA*" has been added in order to test the action of adding a new AC rule to the AC policy. This rule is depicted in **FIGURE 18**. Quentin is a new subject, thus, a new Java object will be generated, containing corresponding capabilities. **FIGURE 19.A** and **FIGURE 19.B** show transmissions between subjects that have access to docA before and after the new rule. As one can notice, Quentin is a new marked subject for docA, proving that the rule has been taken into account in order to maintain the coherence between AC and TC paradigms. Thanks to this test, we valide Challenge **C2**.



(A) (B)

FIGURE 19 – TCL representation of docA before (A) and after (B) the new AC rule. As one can see, a new marked subject has been added for this resource. This subject is able to send and receive the resource to the other marked subjects.

Moving a subject from a SC to another

For this test, a subject is moved from a SubjectCluster to another one, modifying *de facto* her/his capabilities. **FIGURE 20.A** shows that subject Xavier has access to RC_3088906,

and thus, to docB. By moving Xavier to another SC, he is now able to access docA and docX through RC_3088905 (see **FIGURE 20.B**), without being able to access to docB anymore. Moreover, **FIGURE 21** shows the AC rules that have been generated and automatically added to the existing AC policy in order to take into account Xavier's modifications and maintain coherence between AC and TC paradigms. Once again, this test validate the coherence mechanism and Challenge **C2**.

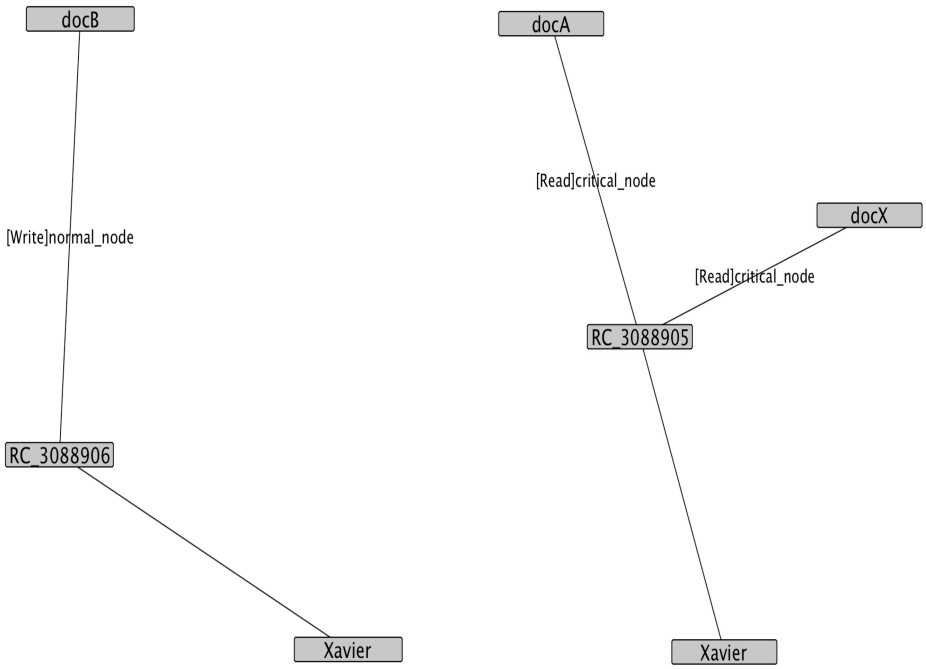


FIGURE 20 – Graphical representation of Xavier's capabilities before (A) and after (B) modification. In (A), Xavier has access to the resource docB (which is in resource cluster RC_3088906). In (B), Xavier's capability has change, he now has access to docA and docX, both members of resource cluster RC_3088905.

Time-consumption overview

In terms of efficiency, Table 3 gives an overview of the different operations that we have tested. As one can see, the coherence mechanism is quite fast (i.e. few seconds), whatever the operation or the ACL size. Indeed, it takes few seconds for most operations to proceed.
Results concerning the coherence mechanism are encouraging for several reasons. First of all, results show that our mechanism could be used in a real-case environment, where AC rules are updated frequently (like our survey has underlined). Secondly, tests show that our model could be used directly from the TC point of view, allowing security experts and administrators to manage their subjects and resources directly from a clustering and capabilities point of view. For instance, an administrator can create a subject by putting her/him in a cluster so he/she can inherit the capabilities of the other members instead of having to define the related Access Control rules. From this perspective, we validate Challenge **C4**.

```
▼<rule name="generated_1">
   <subject subjectName="Xavier"/>
   <action actionName="Read" actionType="non-critical"/>
   <resource resourceLevel="confidential" resourceName="docA" resourceType="image"/>
  </rule>
▼<rule name="generated_2">
   <subject subjectName="Xavier"/>
   <action actionName="Write" actionType="critical"/>
   <resource resourceLevel="confidential" resourceName="docA" resourceType="image"/>
  </rule>
▼<rule name="generated_3">
   <subject subjectName="Xavier"/>
   <action actionName="Read" actionType="non-critical"/>
   <resource resourceLevel="confidential" resourceName="docX" resourceType="image"/>
  </rule>
▼<rule name="generated_4">
   <subject subjectName="Xavier"/>
   <action actionName="Write" actionType="non-critical"/>
   <resource resourceLevel="confidential" resourceName="docX" resourceType="image"/>
  </rule>
```

FIGURE 21 – AC rules that have been automatically generated after Xavier modification.

Action	ACL_3	ACL_7	ACL_11	ACL_15	ACL_20	ACL_24
Create a new subject and add her/him to a specific SC	0,81	1,14	1,69	2,22	1,98	3,19
Create a new subject by giving a set of capabilities	1, 09	1,67	2,80	1,97	1,72	2,74
Modify a subject's capability	0,86	0,21	1,85	0,30	1,12	1,94
Move a subject from a subject cluster to another	1,49	3,24	4,12	1,30	3,72	4,36
Delete an existing subject	0,50	0,23	1,05	0,29	3,07	3,92
Create a new resource and add it to a specific RC	0,90	3,18	2,82	2,09	4,10	5,04
Create a new resource by giving a set of capabilities	0,54	2,37	3,79	3,11	4,29	5,36
Delete an existing resource	0,48	3,84	4,77	4,15	5,03	5,67
Add a new ACL rule	1,34	2,20	2,78	2,23	4,06	4,72
Remove an existing ACL rule	1,28	2,12	3,02	2,98	4,29	5,06

TABLE 3 – Results of the coherence mechanism in seconds.

7.4 Resources similarities

The following subsection describes the tests that we have conducted in order to have insights on resources similarities mechanism. As stated previously, this process aims at tackling the *complexity problem* by creating clusters with similar resources (i.e. resources that can be accessed and retransmitted by the same subjects, in the exact same way). Following tests aim at validating Challenges **C3** and **C4** from the resources similarities perspective.

Data presentation

We have taken the same ACLs that we have used for previous tests to evaluate the resources similarity mechanism in terms of time consumption and efficiency. Empirical tests presented in Table 2 shows interesting results in terms of time consumption. Indeed, results show that most of the time, this process takes less than one hour to process. We underline that this process is only done once (because further modifications will be managed by the coherence mechanism). We can then conclude that this mechanism is quite efficient for the volumetry managed by the participants of our survey. Thus, we validate Challenge **C4**.

The notion of gain (columns 6 and 8 of Table 2) expresses the percentage of reduction of the total number of TCLs. Hypothetically speaking, two extremums can be underlined. In the worst case scenario, every single resource is so different that none of them is accessed in the exact same way by the exact same subjects. In that case, every resource will be in an isolated cluster (i.e. singleton) and every cluster will have a different TCL (there will be n TCLs for n resources).
The opposite extremum is where all resources are the same (i.e. they have identical TCLs) and thus, only one cluster will be generated and will be filled with all resources. In the first extremum, the gain will be of 0%, while the second extremum will show a gain close to a 100%.
Concerning the gain, results show that this mechanism can reduce the total number of TCLs up to 77%. Such results can be interesting, especially in the case of big ACLs that contain thousands of resources (and thus, thousands of TCLs before the resource similarities mechanisms). Thus, we validate Challenge **C3**.

Interpretation

Results show variations of gain and time consumption. Our model represents a TCL as a list of subjects who can access and retransmit one or more resources. Thus, the more subjects in the list, the bigger the TCL. Two TCLs are more likely to be identical if their size (i.e. their list of subjects) is short.
In other words, if for the same amount of subjects and rules, an ACL contains more resources, the gain has better chances to be higher, because the rules will target more resources, and odds to have shorter, and thus, identical TCLs, will be increased.

Now that we have presented the results for the resources similarities mechanism, the following subsection presents the results of the subjects similarities mechanism.

7.5 Subject Similarities

The following subsection describes tests we have been conducted in order to have insights on subjects similarities mechanism. As stated previously, this process aims at creating clusters with similar subjects (i.e. subject that can access and exchange resources in the exact same way). Thanks to these results, we aim at validating Challenges **C3** and **C4** from the subjects similarities perspective.

Raw data presentation

Once again, the same ACLs have been used to do our tests. Concerning the time consumption, one can see that the subjects similarities mechanism is much more faster than the resources similarities mechanism. Indeed, due to the volumetry, the mechanism is taking less than one minute to compute. These results validate Challenge **C4**.
Concerning the gain, one can also notice that the efficiency values are quite variable. Indeed, worst results show a gain of 5% while best results reduces the amount of subjects by 43%. However, we underline that the mean value of the subjects similarities gain is around 20%, which is interesting, especially for infrastructures that contains a lot of subjects. Thus, we validate Challenge **C3**.

Interpretation

The fact that this mechanism is faster than the resource similarities is due to several reasons. First of all, the volumetry of our ACLs. Indeed, where the biggest set of resources embeds 7500 resources, the biggest set of subjects embeds 250 individuals. These differences induce similarities mechanism to be faster.
Moreover, we underline that this mechanism has a lower gain than the resource similarities mechanism. This can be explained by the fact that, one again, the number of subjects is smaller, reducing the odds to have similarities.

In this section, we have presented the tests we have conducted. The objective of these tests was to validate the challenges we have identified thanks to the survey. We have created ACLs that embeds up to 250 subjects and 7500 resources in order to fit participants volumetry.
From the time consumption point of view, results show that our mechanisms can be applied in a real case scenario (and thus validate Challenge **C4**). Indeed, even if the longest process takes more than an hour for biggest ACLs, this process only needs to be done once. Moreover, our algorithms can be optimized by using parallelized code or optimized clustering APIs.
Concerning the efficiency, tests have shown that our resources and subjects similarities mechanism can obtain good results. Of course, results can vary a lot depending on the number of actions [12], the ACL rules, subjects and resources amounts or applied mapping rules (if the rule "*everyone can send and received to/from anyone*", it will be more likely to have similar subjects and resources). However, results show that similarities can be found, even if the TCLs have been filled randomly. Thus, we have validated Challenges **C2** and **C3** as well.

8 Conclusion

Over the years, several Access Control (AC) models have been proposed to tackle the problem of data access within companies. Unfortunately, traditional models do not insure protection over retransmission and data leakage. To overcome this issue, Transmission Control mechanism (TC), such as Data Leak/Loss Prevention (DLP) can be used.
Both AC and TC are based on policies, allowing security experts or administrators to define "*who can access what*" (AC) and "*who can send what to whom*" (TC).

In this paper, we have focused on two objectives. Firstly, we have conducted a survey to gather information on security policy management inside companies (Objective O1). Objective O1 has raised four challenges that we have decided to tackle (Objective O2) :

- Challenge **C1** : the *genericity problem* (i.e. take into account several models that are commonly used in companies),
- Challenge **C2** : the *coherence problem* (i.e. AC and TC that are not coherent with each other),
- Challenge **C3** : the *complexity problem* (i.e. reduce the total number of resources and subjects managed by the policies)
- Challenge **C4** : the *rapidity problem* (i.e. take into account the time-consumption criterion to match the updates frequency of existing AC policies).

12. More actions will generate more capability types, thus reducing the odds to have subjects and resources similarities.

C1 has been resolved by proposing a generic Access Control model to represent commonly used AC models, such as traditional ACL, RBAC or ABAC. **C2** has been answered by defining a generation mechanism to create TC policies that are coherent with the existing AC rules. A coherence mechanisms has also been proposed to keep coherence between AC and TC policies when AC and/or TC rules are modified. Furthermore, **C3** has been resolved by providing inference mechanisms that clusterize subjects and resources based on similarities, reducing the tiresomeness of management. Finally, **C4** has been answered empirically thanks to tests we have been conducted.

For future works, we first want to focus our efforts on Human-Machine Interactions (HMI) by offering mechanisms that would help enhancing the overall knowledge of security experts and administrators. These mechanisms have been considered as interesting by the participants of our survey. Moreover, we want to take a formal approach regarding the complexity of our algorithms in order to improve their efficiency and time-consumption. Finally, we want to test our solution on real Access Control policies.

Acknowledgment

This work has been realized under the FUI (Fond Unique Interministériel) 4TRAX.

Bibliographie

[1] Tcsec, D. O. D. "Trusted computer system evaluation criteria." DoD 5200.28-STD 83 (1985). 4

[2] Saltzer, J.H., Schroeder, M.D. : The protection of information in computer systems. Proceedings of the IEEE , vol.63, no.9, pp.1278,1308, Sept. (1975) doi : 10.1109/PROC.1975.9939 (1975) 4

[3] Graham, G. Scott, and Peter J. Denning. "Protection : principles and practice." Proceedings of the May 16-18, 1972, spring joint computer conference. ACM, (1972). 4

[4] Sandhu, Ravi S., et al. Role-based access control models. Computer 2 : 38-47. (1996) 5

[5] Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K. Guide to attribute based access control (ABAC) definition and considerations. NIST Special Publication, 800, 162. (2014) 5

[6] Han, W., & Lei, C. A survey on policy languages in network and security management. Computer Networks, 56(1), 477-489. (2012) 5

[7] Jack B. Dennis and Earl C. Van Horn, Programming Semantics for Multiprogrammed Computations, Communications of the ACM, Volume 9, Issue 3, March 1966, pp. 143 ?155, DOI=10.1145/365230.365252. (1966) 5

[8] Fabry, Robert S. "Capability-based addressing." Communications of the ACM 17.7 (1974) : 403-412. (1974) 5

[9] Jones, Anita K., et al. "StarOS, a multiprocessor operating system for the support of task forces." Proceedings of the seventh ACM symposium on Operating systems principles. ACM, 1979.multics (1979)

[10] elimi, Mennan, and Felix Freitag. "Tahoe-LAFS Distributed Storage Service in Community Network Clouds." Big Data and Cloud Computing (BdCloud), 2014 IEEE Fourth International Conference on. IEEE, (2014) 5

[11] Miller, Mark S., et al. Safe active content in sanitized JavaScript. Technical report, Tech. Rep., Google, Inc, (2008) 5

[12] Mettler, Adrian, David Wagner, and Tyler Close. "Joe-E : A Security-Oriented Subset of Java." NDSS. Vol. 10. (2010) 5

[13] Miller, Mark S., Ka-Ping Yee, and Jonathan Shapiro. Capability myths demolished. Technical Report SRL2003-02, Johns Hopkins University Systems Research Laboratory, 2003. http ://www. erights. org/elib/capability/duals, (2003) 5

[14] Bell, D. "The bell-lapadula model." Journal of computer security 4.2 (1996) : 3. 5

[15] Biba, Kenneth J. Integrity considerations for secure computer systems. No. MTR-3153-REV-1. MITRE CORP BEDFORD MA, (1977) 5

[16] Corbato, Fernando J., and Victor A. Vyssotsky. "Introduction and overview of the Multics system." Proceedings of the November 30 December 1, 1965, fall joint computer conference, part I. ACM, (1965) 5

[17] Clark, David D., and David R. Wilson. "A comparison of commercial and military computer security policies." Security and Privacy, 1987 IEEE Symposium on. (1987) 5

[18] Shabtai, A., Elovici, Y., & Rokach, L. (2012). A survey of data leakage detection and prevention solutions. Springer Science & Business Media. (2012) 6

[19] Gafny, Ma'ayan, et al. "Detecting data misuse by applying context-based data linkage." Proceedings of the 2010 ACM workshop on Insider threats. ACM, (2010) 6

[20] Li, Hao, et al. "Leakage Prevention Method for Unstructured Data Based on Classification." Applications and Techniques in Information Security. Springer Berlin Heidelberg, 2015. 337-343. (2015) 6

[21] Chae, Cheol-Joo, et al. "A privacy data leakage prevention method in P2P networks." Peer-to-Peer Networking and Applications (2015) : 1-12. 6

[22] Zilberman, Polina, et al. "Analyzing group communication for preventing data leakage via email." Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on. IEEE, (2011) 6

[23] Alawneh, Muntaha, and Imad M. Abbadi. "Preventing information leakage between collaborating organisations." Proceedings of the 10th international Conference on Electronic Commerce. ACM, (2008) 6

[24] Wuchner, Tobias, and Alexander Pretschner. "Data loss prevention based on data-driven usage control." Software Reliability Engineering (ISSRE), 2012 IEEE 23rd International Symposium on. IEEE, (2012) 6

[25] Kelbert, Florian, and Alexander Pretschner. "Data usage control enforcement in distributed systems." Proceedings of the third ACM conference on Data and application security and privacy. ACM, (2013) 6

[26] Park, Jaehong, and Ravi Sandhu. The UCON ABC usage control model. ACM Transactions on Information and System Security (TISSEC) 7.1 : 128-174. (2004) 6

[27] M. Hilty et al., A Policy Language for Distributed Usage Control Proc. European Symp. Research in Computer Security, Springer-Verlag, 2007, pp. 531 ?546. (2007) 7

[28] Gheorghe, G., Mori, P., Crispo, B., Martinelli, F. : Enforcing UCON Policies on the Enterprise Service Bus. In : OTM Conferences (2). pp. 876-893 (2010) 7

[29] Cuppens, F., Cuppens-Boulahia, N., & Ghorbel, M. B. (2007). High level conflict management strategies in advanced access control models. Electronic Notes in Theoretical Computer Science, 186, 3-26. (2007) 7

[30] Ayed, Samiha, Nora Cuppens-Boulahia, and Frederic Cuppens. Managing access and flow control requirements in distributed workflows. Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on. IEEE, (2008) 7

[31] Ayed, Samiha, Nora Cuppens-Boulahia, and Frederic Cuppens. An integrated model for access control and information flow requirements. Advances in Computer Science ASIAN 2007. Computer and Network Security. Springer Berlin Heidelberg, 2007111-125. (2007) 7

[32] Johnson, Kevin, and Barbara L. Filkins. "SANS Mobility/BYOD Security Survey." SANS Institute Whitepaper. https://www.sans.org/reading-room/whitepapers/analyst/cybersecurity-professional-trends-survey-34615 (2012) 7

[33] Cryptzone, Network Security Survey Results : Is Network Access Putting You at Risk ? https://www.cryptzone.com/pdfs/Whitepapers/Cryptzone-Network-Access-Security-Survey-2015.pdf (2015) 7

[34] SANS Institute InfoSec Reading Room, Cybersecurity Professional Trends : A SANS Survey (2014) 7

[35] Yoann Bertrand, Mireille Blay-Fornarino, Karima Boudaoud, Michel Riveill : Generation of Transmission Control Rules Compliant with Existing Access Control Policies. SecureComm 2015 : 438-455 (2015) 13

[36] Barker, S. (2012). Logical approaches to authorization policies. In Logic Programs, Norms and Action (pp. 349-373). Springer Berlin Heidelberg. (2012) 14

[37] Slimani, Nadera, et al. UACML : Unified access control modeling language. New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on. IEEE, (2011) 14

[38] Khamadja, S., Adi, K., & Logrippo, L. (2013, November). An access control framework for hybrid policies. In Proceedings of the 6th International Conference on Security of Information and Networks (pp. 282-286). ACM. (2013) 14