Too Long, did not Enforce: A Qualitative Hierarchical Risk-Aware Data Usage Control Model for Complex Policies in Distributed Environments

Fabio Martinelli, Christina Michailidou, Paolo Mori and Andrea Saracino
Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche
Pisa, Italy
name.surname@iit.cnr.it

ABSTRACT
Distributed environments such as Internet of Things, have an increasing need of introducing access and usage control mechanisms, to manage the rights to perform specific operations and regulate the access to the plethora of information daily generated by these devices. Defining policies which are specific to these distributed environments could be a challenging and tedious task, mainly due to the large set of attributes that should be considered, hence the upcoming of unforeseen conflicts or undesired conditions. In this paper we propose a qualitative risk-based usage control model, aimed at enabling a framework where is possible to define and enforce policies at different levels of granularity. In particular, the proposed framework exploits the Analytic Hierarchy Process (AHP) to coalesce the risk value assigned to different attributes in relation to a specific operation, in a single risk value, to be used as unique attribute of usage control policies. Two sets of experiments that show the benefits both in policy definition and in performance, validate the proposed model, demonstrating the equivalence of enforcement among standard policies and the derived single-attributed policies.

CCS CONCEPTS
• Security and privacy → Access control;

KEYWORDS
Risk analysis; Usage Control; Analytic Hierarchy Process

ACM Reference Format:

1 INTRODUCTION
Controlling the right to access data and perform operations has become a challenging task during the last years. In fact, data are continuously produced and shared by an always increasing number of applications, such as social networks, business applications, cloud environments etc. Considering the distributiveness of today’s applications and environments and also the dependence of access policies from context conditions and its rapid and dynamic changes, a flexible and configurable control system is needed. This becomes even more relevant in the Internet of Things (IoT) paradigm, which groups a plethora of applications, environments and settings where a set of interconnected devices create, exchange and manipulate data, taking decisions autonomously, which generally have a direct effect on the physical world and people lives.

The most widely used access control models, such as the Discretionary Access Control (DAC) [4], the Mandatory Access Control (MAC) [4], the Role Based Access Control (RBAC) [20] and the Attribute Based Access Control (ABAC) [6] are common models used to express data access policies. However, all these models do not address two relevant issues: (i) continuity of access and (ii) policy complexity. More specifically performed operations and data access are not always instantaneous, e.g. accessing a file from a text editor, data streaming, virtual machines use, machine operations etc. Hence, the right to perform a specific operation should be monitored and verified also during usage, in case the context mutates and the policy is not matched anymore. Concerning (ii) dynamic environments such as IoT require that in policies a large number of attributes are considered, which makes the policy writing process complex and tedious, also increasing the probability of inserting inconsistency and mistakes.

Usage control model (UCON) [15] comes to fill the first gap (i), by improving traditional ABAC introducing two main novelties: the consideration of mutability of attributes and the continuity of access decision evaluation. UCON can basically exploit any kind of attribute for policy evaluation. Attributes are related to the subject wishing to perform the access, to the resource to be accessed and to the context in which the access is performed. These attributes are generally qualitative and quantitative values on which rules are defined with conditions of equality or comparison. Complex settings will then require quite complex policies, where allowed values might be specified for each considered attribute. In the process of simplifying the policies, two possible strategies are reducing the amount of considered attributes, or coalescing (aggregating) more values in a single one. The first approach would require a prioritization of the attributes and causes a lost of expressiveness, since several conditions cannot be expressed. The aggregation, though it also brings an intrinsic reduction of expressiveness, still considers all of the original attributes, since the aggregated value results from a computation which takes all the original attributes as an input. In this direction, it is possible to assign a level of risk to specific
settings related to the operation or resource that should be controlled. For example if the access policy considers the attribute of location, to allow or deny the access to a resource, it is possible to identify different levels of risks to be assigned to different locations, e.g. low risk if the access request is performed nearby the accessed resource, high risk if performed far from it. Thus, if a level of risk for specific operation is assigned to different attributes for their values, it is possible to aggregate these attributes through a uniform risk measurement, to be used as a value for expressing policy.

The current state of the art, usually combines the notion of risk with the trust level of the requesting subject. Mainly in the RBAC model, risk is a variable which depends on how the subject uses the given privileges during the access [1][21][22]. Other studies examine the tradeoff between the benefits of allowing the access and the cost of denying access [3][23][9]. Also, in these studies risk refers to the subject’s trustworthiness and object’s sensitivity.

In this paper we propose a qualitative risk-based model for usage control, configurable and flexible designed to simplify the process of writing risk-based policies, coalescing multiple attributes in a single value, which describes the risk level of performing a security critical operation. The proposed framework exploits the Analytic Hierarchy Process (AHP), to aggregate multiple attributes of a single request, in a single aggregated value, simplifying thus the effort of policy writing and evaluation. In this framework we assume that the attributes which might be used to define and evaluate UCON policies, come with an associated level of risk. Thus, our model aggregates different attributes through the definition of an AHP problem, which is aimed at assessing the specific risk level of a specific request. In particular, the risk level, which is expressed in a qualitative manner, will provide the possible solutions to the AHP problem, and the attribute values will have different relevance to move the AHP decision toward a specific risk level. Hence, we exploit AHP to calculate the relevance of each attribute with respect to the total risk of a request, getting as output one single value, which will characterize the aggregated level of risk of such a request. This allows to define simplified policies with a single attribute, where the framework handles the access/usage requests, by automatically translating them to the required granularity level, specified by the policy type. In particular, the proposed framework can be scaled to define a hierarchy of subproblems, where a specific risk value for a subproblem can be obtained by aggregating a subset of related attributes, e.g. “temperature” and “geolocation” define the risk level related to “environment”. Then, the risk values of subproblems can be aggregated by an additional AHP decision process to define a single final risk level. This allows the definition of policies with different levels of granularity, making the proposed model more flexible and suitable to representation of complex and distributed environments.

The paper will present a Reverse Tree-based model for attribute aggregation, and the correspondent usage control framework, based on a clustered structure for attribute retrieval, which allows the enforcement of the risk based policies at different levels of granularity. Finally, two sets of experiments will be presented. One comparing the evaluation of requests with a full set of attributes and the derived risk-based policies and one showing the performance analysis of the proposed model.

The rest of the paper is organized as follows. In Section 2, we present a small background both for the Usage Control model and the Analytic Hierarchy Process. Section 3 is devoted to the presentation of the proposed model and its functionality and also the integration of the framework in UCON is presented. In Section 4 a use case is described to give a better understanding of the model’s functionality. In Section 5 two sets of experiments and the corresponding results are presented. Section 6 describes the relevant works in the related fields, which are compared with the proposed model. Finally, Section 7 concludes the paper.

2 BACKGROUND

In this section we will briefly review the theoretical background of the Usage Control and of the Analytic Hierarchy Process, which are exploited in this paper.

2.1 Usage Control

The Usage Control (UCON) model [16, 18] extends traditional access control models introducing mutable attributes and new decision factors besides authorizations, obligations and conditions. Mutable attributes represent features of subjects, resources, and environment that change their values as a consequence of the normal operation of the system [17]. For instance, some mutable attributes change their values because the policy includes attribute update statements that are executed before (pre-update), during (on-update), or after (post-update) the execution of the access. For instance, the e-wallet balance is a subject attribute which could be decreased by the policy every time the subject performs a new access to a resource.

Since mutable attributes change their values during the usage of an object, the usage control model allows to define policies which are evaluated before (pre-decision) and continuously during the access to the object (ongoing-decision).

The continuous evaluation of the policy when the access is in progress is aimed at executing proper countermeasures (such as interrupting the access) when the execution right is no more valid, in order to reduce the risk of misuse of resources. Hence, in the Usage Control model it is crucial to be able to continuously retrieve the updated values of the mutable attributes, in order to perform the continuous evaluation of the policy and to promptly react to the attribute change by taking proper actions, e.g., by interrupting those ongoing accesses which are no longer authorized.

This paper takes into account Usage Control systems based on the XACML reference architecture, with particular reference to the one we presented in [2, 13], which is shown in Figure 1. In the XACML reference architecture, the Policy Enforcement Points (PEPs) embedded in the controlled system intercept the execution of security relevant operations, and they invoke the Context Handler (CH), which is the frontend of the Usage Control system. The Policy Information Points (PIPs) are the components invoked by the CH to retrieve the attributes required by the Policy Decision Point (PDP) for the execution of the decision process, i.e., to evaluate the policy retrieved from the Policy Store (PS). Attributes are managed by Attribute Managers (AMs), sometimes called Attribute Providers or Attribute Stores, which provide the interfaces to retrieve and, in case of mutable Attributes, to update their current values. Each
specific scenario where the Usage Control system is exploited requires its own set of AMs to manage the attributes required for the policy evaluation. Hence, PIPs are properly configured in order to be able to query the specific AMs adopted in the scenario of interest for retrieving and updating attributes. In particular, each PIP implements the specific protocol required to interact with the related AM and exploits the provided mechanisms for securing the communications. The Usage Control model emphasizes the role of PIPs because it introduces the continuous policy enforcement while an access is in progress to cope with mutable attributes. In particular, the PIP is also in charge of detecting attribute changes in order to trigger the policy re-evaluation for the involved ongoing accesses, which are managed by the Session Manager (SM). To detect attribute changes, the PIP could exploit the subscription mechanism provided by the AM or the PIP must emulate it if it is not supported by the AM.

The phases of the Usage Control decision process are regulated by the interactions between the PEP and the Usage Control systems as follows (derived from [24]):

1. **TryAccess**: is the pre-decision phase, which begins when the TryAccess message is sent by the PEP to the Usage Control system because a subject requests to execute the access. The TryAccess phase finishes when the Usage Control system sends the response to the PEP. The possible responses are: PERMIT, to allow the access, or DENY;

2. **StartAccess**: is the first part of the ongoing-decision phase, which begins when the StartAccess message is sent by the PEP to the Usage Control system because the access has just started, and finishes when the policy has been evaluated and the response has been sent back to the PEP;

3. **RevokeAccess**: is the second part of the ongoing-decision phase. This phase is executed every time an attribute changes its value. This phase starts when an attribute changes its value. It finishes when the policy has been evaluate and, if a policy violation occurs, the RevokeAccess message is sent by the Usage Control system to the PEP.

### 2.2 Analytic Hierarchy Process

Analytic Hierarchy Process (AHP) is a widely used method for modeling and decomposing into an hierarchy of subproblems. Multi Criteria Decision Making (MCDM) problems. It was first introduced by Saaty in [19] and finds application in many fields, from finance to crime investigation. One of the biggest advantages of AHP is that it gives the opportunity to the decision maker to combine qualitative and quantitative values. For example, with AHP the coexistence of values that represent a weight or a price with values that represent feelings or preferences is possible. AHP method is based on three main elements: the goal, the criteria and the alternatives.

**Goal**: It is the reason why this decision procedure is being conducted. The goal represents the final outcome. For example, a goal could be to select the best place for summer vacation.

**Criteria**: These are the factors on which the decision is based. In the previous example, the criteria could be the place, the price, the room availability etc.

**Alternatives**: This element of the hierarchy is basically the choices that we have in our disposal. One of these alternatives will be the final solution of the initial problem. The alternatives for the vacation decision could be countries or cities.

The AHP method can be divided into steps. The first one is the construction of the pairwise comparison matrices, where the decision maker can give a level of preference to the criteria and the second is the computation of the priority vectors.

#### 2.2.1 Pairwise Comparison Matrices

For each criterion the decision maker describes through a comparison matrix the relevance of an alternative over the others with respect to this criterion. Furthermore, a comparison matrix where the relevance of each criterion with respect to the goal is designated, is also constructed for the final decision. A comparison matrix is a square $n \times n$ matrix, where $n$ is the number of the alternatives. Each element $a_{ij}$ of the matrix takes a value in the $[1, ..., 9]$ interval which defines how important this element is considered to be in comparison with another one. The main characteristic of the comparison matrices is that they are reciprocal. Thus, if for instance the alternative $i$ is $3$ times more relevant than the alternative $j$, then $a_{ij}$ equals to $3$ and $a_{ji}$ equals to $\frac{1}{3}$. So, while filling this matrix a general rule is that $a_{ij} = \frac{1}{a_{ji}}$.

#### 2.2.2 Computing the priority vectors

The priority vectors show the relative weights among the alternatives we compare. For the computation of these vectors the normalization of the comparison matrices is needed, Formulas 1 and 2.

$$
Sum_j = \sum_{j=1}^{n} a_{ij}
$$

$$
\bar{a}_{ij} = \frac{a_{ij}}{Sum_j}
$$

Having the normalized values the priority vector or also called the principal eigen vector can be calculated using the Formula 3.

$$
\omega_j = \frac{1}{n} \sum_{i=1}^{n} a_{ij}
$$
Except for the relative weights, the evaluation of the consistency is also needed. Due to the subjectivity that the AHP method encapsulates, inconsistencies coming from the declaration of the comparison matrices may be present. To check if the model is consistent and valid the Consistency Index (CI) must be calculated according to Formulas 4 and 5. As claimed by [19] the CI value must be compared with the appropriate one which is called Random Consistency Index (RI). For the comparison the Consistency Ratio is used, Formula 6. If the outcome of the comparison gives a consistency ratio under 10% that implies that the inconsistency is in an acceptable level.

\[ \lambda_{\text{max}} = \sum_j \text{Sum}_j w_j \]  \hspace{1cm} \text{(4)}

\[ CI = \frac{\lambda_{\text{max}} - N}{N - 1} \]  \hspace{1cm} \text{(5)}

\[ CR = \frac{CI}{RI} \]  \hspace{1cm} \text{(6)}

Despite the static AHP, where for each criterion a comparison matrix is constructed to describe the relevance of the alternatives with respect to it, there is also the Dynamic AHP which will be used for the purposes of this study. In the Dynamic AHP a number of comparison matrices can be available for each criterion. Based on the value of the criterion the corresponding matrix is chosen and the model adjust to the current context. An application of the Dynamic AHP can be found in [5], where the authors used this method to characterize the risk level of an Android application, helping the user decide whether to install it or not. Figure 2 depicts the aforementioned model, where for a Criterion X there are N comparison matrices, each one related to the possible value of X. Hence, according to the value of X the corresponding matrix is chosen and used for the rest of the procedure.

\[ \lambda_{\text{max}} = \sum_j \text{Sum}_j w_j \]  \hspace{1cm} \text{(4)}

\[ CI = \frac{\lambda_{\text{max}} - N}{N - 1} \]  \hspace{1cm} \text{(5)}

\[ CR = \frac{CI}{RI} \]  \hspace{1cm} \text{(6)}

Figure 2: Dynamic Analytic Hierarchy Process

3 RISK MODEL AND FRAMEWORK
DESCRIPTION

In an attribute-based access control model, when the subject requests access to an object, a set of attributes must be acquired in order to check if the security policy is satisfied. Most of the times, these attributes come from different sources. For example, the request may take into account the role of the subject, its trustworthiness, environmental factors etc. All attributes, depending on the values they have, can give to the request a certain level of risk. For instance, a request made by an employee to access a sensitive resource from a public network is far riskier than the same request made by the administrator while connected to the company’s network.

The proposed model is a qualitative risk model for systems that exploit Usage Control and its main goal is to characterize the total risk related to the execution of a given access/usage request related to the current access context. This model takes into account both the risk that comes with each attribute, based on its value, and the importance that the attribute has for the decision-making. The model exploits the AHP method and it can be adjusted to the requirements of the application environment.

3.1 Risk Computation for UCON Request

The proposed framework aims at simplifying the process of policy evaluation by aggregating the attributes which are describing the current context, in a smaller and more manageable set of attributes. This single value is calculated by exploiting the risk value assigned to each attribute which participate in the access context, combining them afterward, by means of a multi-criteria decision process. The terms used throughout the rest of the paper are defined as follows

- **Full Policy**: A policy considering the attributes as they are extracted from AMs, hence not aggregated.
- **RA-Policy**: A risk aware policy is a policy which is written by considering the risk level of aggregated attributes. It has generally a smaller number of attributes with respect to the correspondent Full-Policy, hence it is easier to define and evaluate.
- **Initial Request**: A request generated by the PEP and enriched with the attributes extracted by the PIPs connected to AMs.
- **Aggregated Request**: Is a request automatically computed by our framework, starting from an initial request, translating it to the aggregation level required by the current RA-Policy.

**Workflow**: In the standard workflow of the proposed framework (depicted in Figure 3), we suppose that the policies are defined as RA-Policies, to simplify the effort of the policy writer and stored in the PAP of the UCS. The defined policies is oblivious to the PEP which generates the initial request, generally only specifying the basic attributes needed to identify subject and object of the request. The initial request is then enriched by the PIP chain, as it happens for the standard UCON workflow. However, in the proposed framework the request will be enriched with the standard attributes coming both from the standard AMs, and from the AHP decision processes that will add the risk-based attribute values. Hence, the Full Request will include both the standard attributes and the risk-based ones. Finally, the PDP will evaluate the request against the stored RA-policy, which only includes risk-based attributes, hence the standard attributes of the Full Request will not be considered, reducing thus the computational overhead of the framework.

The proposed model is based on a Total Risk Reverse Tree representation, which is shown in Figure 4. The root of this tree represents the total risk value obtained by combining the attributes extracted
Figure 3: Operative workflow of the proposed framework.

Figure 4: Total Risk Reverse Tree

from the upper level (Level A). Thus, considering for the sake of simplicity that there are no other levels, the values in Level A will define the risk assigned to the single attributes of the UCON access context (i.e. the Full Policy), whilst the Total Risk represents the global value of risk assigned to the request, which should be considered in the simplified policy, the RA-Policy. This aggregation process can be scaled at different levels, by grouping, for example, related attributes that contribute to a specific risk aspect. This is done by adding an additional level to the reverse-tree. Hence, the leaves (top-level) are the attributes considered in the Full Policy, the Total Risk is the single attribute to be used in the RA-Policy, whilst the values at intermediate levels represent the risk value assigned to aggregated set of attributes. This allows the definition of policies at different levels of granularity.

Example: Let’s consider the following usage control policies: “File A can be read by a user on a mobile phone if he is an administrator, is authenticated on the device, is using less than five opened applications, he is in the workplace premises and is connected to the company WiFi network”. It can be noted that the policy considers five attributes, both qualitative and quantitative, being thus a Full Policy. The action to be controlled is the usage of a resource which is the File A, which should not be allowed unless the user is an administrator and the access is regulated by these five conditions concerning device integrity, user authentication and geolocation. The fact that the user is in the company premises and that is connected to the company’s WiFi are related to the risk of performing the access in a non-controlled environment, whose value is “low” should the access be performed in the company premises (monitored through the geolocation attribute). Hence, these two attributes can coalesce in a single risk value, which will be “low” when the conditions expressed for these two attributes, match the policy requirements. Following this schema, it is possible to aggregate also the remaining attributes, extracting thus a final total risk value. The RA-Policy for such an attribute will thus become: “File A can be read by a user on a mobile phone if the current risk level is lesser or equal than Low”.

As anticipated, the methodology used to aggregate attributes is the AHP decision model, where the possible alternatives are the possible risk levels (e.g. Low Risk, Medium Risk, High Risk) and the criteria are the attributes, used to identify as a goal the level of risk (among the alternatives) of performing a specific operation in the current context. More specifically, our model defines for each attribute value its relevance to the global risk level, stating how it moves the decision toward each specific alternative. Hence, for each value (or range of values) of each attribute, is defined a comparison matrix to be used in a specific instance of an AHP decision problem. This dynamic AHP model has already been used in [5], and allows the definition of different decisions problems, whose structure depends and adapts to the current context. The comparison matrices are defined according to the number of risk level which have to be considered for a specific problem, and the values are assigned according to the recommendation of an expert with a deep knowledge of the specific environment where the access/usage has to be controlled.

3.2 Integration of the risk-aware framework in UCON model

To introduce the proposed qualitative risk model in the UCON architecture, no formal extensions are required. The evaluation of risk-based policies, in fact, only requires the presence of attributes expressing the level of risk. Hence, the process of aggregating attributes through the AHP is completely oblivious to the UCON model, which has to be modified only by defining a set of PIPs to read the risk level attributes from each AHP block in the reverse tree of Figure 4. The proposed architecture is depicted in Figure 5. As shown, the Usage Control System (UCS) architecture is not modified, still, the inclusion of risk based attributes, is represented by the AHP blocks, which act as Attribute Managers (AMs). The AHP blocks read the attribute values from standard AMs and dynamically computes the risk value, which is queried by the additional PIPs. More specifically, in Figure 5 the reported example implements a two-level risk computation problem, where the first set of two attributes (red box) are aggregated, i.e. act as criteria for an AHP single level problem, and the second set of four attributes (green box) are the criteria for a second AHP single level problem.
The two decisions, are then further aggregated in another AHP problem which will compute the Total Risk level, aggregating thus all attributes in a single value.

It is worth noting that, this architectural model allows the definition of policies of any granularity level. In fact, it is possible to define policies using attribute values from standard AMs, policies with one or two risk attributes at the first level, or policies considering only the aggregated risk value. Moreover, it is possible to define hybrid policies considering attributes from different level. For example, it is possible to define a policy which specifies the accepted Total Risk level of performing an operation, still defines a specific constraint on a specific attribute, e.g. “The user can access the file $A$ if the Total Risk level is not higher than Medium and the user role is Administrator”. Depending on the granularity level specified in the policy, the proposed framework will automatically adapt the original request, by adding the attributes extracted through the PIPs connected to the AHP AMs. Hence, the proposed architecture proves to be general and easily configurable, allowing the description of problems and the related policies with any level of complexity and attribute aggregation strategies, completely configurable granularity and without modifying the standard UCON implementation model discussed in Section 2.

4 USE CASE

In this use case, we consider an IoT system integrated in a company, whose functionalities are handled through configuration files stored in a cloud system. We suppose that an employee (subject), wishes to access and modify a system configuration document (object), from a mobile device, such as a smartphone or tablet. This operation should be allowed only in safe conditions, since unwanted and unauthorized access can cause disclosure of privacy sensitive information and system misconfiguration which might result in physical damages to the system or involved people. To model the context in which the access request is performed, we assume that the attributes are related to three semantic groups, namely Environment, System and Subject Trust Level. Figure 6, schematically depicts the sources and the attributes for each attribute group which are detailed in the following. It is worth noting, that the attributes of the proposed example are mutable, hence access control will not suffice, requiring thus the definition of usage control policies. Based on these attributes, a set of Full Policies has been defined, which regulate the right to access a specific documents. Afterward a correspondent set of RA-policies has been defined, to evaluate the capability of the framework to cope with the reduced expressiveness of the aggregated attributes. The policies have been matched with a set of Initial Requests, representing the attributes values from five different scenarios that we have defined.
4.1 Risk level computation

Each attribute encapsulates a level of risk, depending on the values that can take at the time of the request or during the access. The attributes we are considering are the following and a summary of them can be found in Table 1:

- **Time**: Expresses the moment at which the access request is issued. It is expressed as a three-valued string whose possible values are Working Hours, Lunch Break, Not At Work, representing respectively the time ranges 8:00 - 12:00, 12:01 - 14:00 and 14:01 - 17:00.
- **Location**: Is the device geolocation from which the access request is issued. The possible values are In premises, Near Premises, Far from premises, directly converted from the GPS coordinates, according to configured circular areas, around the working place.
- **Network**: Is the network in which the device is connected, the possible values are Company’s Network/WiFi, Network with Encryption (i.e 3G/4G), Public Network.
- **Installed Applications**: This attribute refers to the trust value of the applications that are installed in the device extracted through the method described in [5], the possible values are Trusted, Dangerous, Critical
- **Antivirus**: This attribute refers to the presence of an antivirus in the device, the possible values are Present, Outdated antivirus, No antivirus

For the attribute of Time we assume that, if it is between 8:00 - 12:00 or 2:01- 5:00 (Working hours, excluding lunch break) the level of risk is Low; if it is between 12:01-2:00 or 5:01-7:00 (Lunch break, close to the end of the work shift) the risk level is Medium and finally after 7:00 the risk level is High. The rationale of this risk assignment is that it is less likely that a document related to work will be accessed out of working hours. For what concerns Location, if the access is performed while the user is In premises of the company/organization then the risk level is Low, if it is Near premises (within 0.5 km of the premises) is Medium and if it is Far from premises is High. For the network attribute, we suppose that if the subject is connected to the Company’s network the level of risk is Low, if it is connected to a Network with encryption (i.e 3G/4G) the risk is Medium and if the network is Public network the risk is High since the device can easily be compromised by other malicious users of the network. The antivirus attribute comes with a Low level of risk if there is an antivirus Present in the device, because we can expect that the device is protected from viruses and other malicious programs, with a Medium risk if the antivirus is Outdated and with a High level of risk if there is No antivirus program installed. The last attribute refers to the applications that are installed in the device. Some of these applications may use permissions that can be risky during the access of sensitive information, such as permission to access stored documents. Moreover, there are also applications which are called mock location applications and are designed to cover the real location of the user. Hence, for this attribute we claim that the level of risk is Low if there are Trusted applications installed, it is Medium if there are applications which have been characterized as Dangerous and it is High if there are Critical applications installed, such as the mock location. Finally, as we already stated, the proposed model is configurable and can use many and different methods to calculate the risk value of each block. In this example, we assume that the risk value coming from the Subject Trust Level block is being calculated not with the AHP method but with another procedure which is out of the scope of this paper.

We recall that for each possible attribute value a different comparison matrix is defined, to be used by the specific instance of the AHP problems. An example of the comparison matrices that were used in the AHP procedure is shown in Tables 2,3,4.

### Table 1: Summary Of The Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Values</th>
<th>Type (Discrete/Continuous)</th>
<th>Mutable (Yes/No)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Working Hours, Lunch Break, Not At Work</td>
<td>Discrete</td>
<td>Yes</td>
</tr>
<tr>
<td>Location</td>
<td>In premises, Near Premises, Far From premises</td>
<td>Discrete</td>
<td>Yes</td>
</tr>
<tr>
<td>Network</td>
<td>Company’s Network/WiFi, Network with Encryption, Public Network</td>
<td>Discrete</td>
<td>Yes</td>
</tr>
<tr>
<td>Installed Apps</td>
<td>Trusted, Dangerous, Critical</td>
<td>Discrete</td>
<td>Yes</td>
</tr>
<tr>
<td>Antivirus</td>
<td>Present, Outdated antivirus, No antivirus</td>
<td>Discrete</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Table 2: Comparison Matrix of the alternatives for working hours

<table>
<thead>
<tr>
<th>Working Hours</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Unacceptable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>1</td>
<td>7</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Medium</td>
<td>1/7</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>High</td>
<td>1/9</td>
<td>1/3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Unacceptable</td>
<td>1/9</td>
<td>1/3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 3: Comparison Matrix of the alternatives in case of public network

<table>
<thead>
<tr>
<th>Public Network</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
<th>Unacceptable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>1</td>
<td>1/4</td>
<td>1/9</td>
<td>1/9</td>
</tr>
<tr>
<td>Medium</td>
<td>4</td>
<td>1</td>
<td>1/9</td>
<td>1/9</td>
</tr>
<tr>
<td>High</td>
<td>9</td>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Unacceptable</td>
<td>9</td>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 4: Comparison Matrix of the alternatives in case of public network

<table>
<thead>
<tr>
<th>Level</th>
<th>Environment</th>
<th>System</th>
<th>Trust Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Medium</td>
<td>4</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>High</td>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Level</th>
<th>Environment</th>
<th>System</th>
<th>Trust Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>1</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Medium</td>
<td>4</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>High</td>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
5 EXPERIMENTAL EVALUATION
To evaluate the proposed model we conducted two sets of experiments. The first one aims to evaluate the model in terms of the final decision, comparing the decision that UCON would have made taking into account the Full Policy and the Initial Request and the decision that the RA-UCON made based on the RA-Policy and the Aggregated Request. The second experiment has been made to validate the proposed model in terms of performance, examining the time needed by a PDP to evaluate a Full Policy and a RA-Policy and return the final decision.

5.1 Policy Evaluation
In order to conduct the first experiment, we defined a set of Full Policies, the corresponding RA-Policies and a number of scenarios, which are shown below.

FP1: Allow access to document if user is in Company’s Premises, connected to Company’s WiFi and device has installed only Trusted applications, else deny.

RAP1: Allow access to document if the total risk of the request is Low, else deny.

FP2: Allow access to document if the time is in the Working Hours, the user is in the Company’s Premises or Near the premises and an antivirus program is Present in the device, else deny.

RAP2: Allow access to document if the the total risk of the request is at most Low and the time is in the Working Hours, else deny.

FP3: Allow access to document if the time is in the Working Hours or during the Lunch Break and the user is in the Company’s Premises or Near the premises, else deny.

RAP3: Allow access to document if the the total risk of the request is at most Medium, else deny.

FP4: Allow access to document if the time is in the Working Hours, the user is in the Company’s Premises, an antivirus program is Present, device has installed only Trusted applications and it is connected to the Company’s WiFi, else deny. The risk level of the user must not be higher than medium.

RAP4: Allow access to document if the the total risk of the request is Low and the risk of trust manager block is at most medium, else deny.

FP5: Allow access to document if the time is in the Working Hours or during the Lunch Break, the user is in the Company’s Premises or Near the premises, an antivirus is installed even outdated, device has installed only Trusted applications and it is connected at least to a network with WPA encryption, else deny. The risk level of the user must be low.

RAP5: Allow access to document if the the total risk of the request is Low, else deny.

Such policies will be verified against a set of possible situations-scenarios:

S1: User is in Company’s Premises, in the Working Hours, connected to a network with WPA encryption, does not have antivirus and has 8 Dangerous apps. Risk level of the user regarding the trust attribute is medium.

S2: User is Near the premises, in the Working Hours, connected to Public Network, an antivirus is Present and the device has installed only Trusted applications. Risk level of the user regarding the trust attribute is medium.

S3: User is Far from premises, Not at work, connected to a network with WPA encryption, an antivirus is Present and the device has installed only Trusted applications. Risk level of the user regarding the trust attribute is high.

S4: User is in Company’s Premises, during the Lunch Break, connected to the Company’s WiFi, does have an Outdated Antivirus and the device has installed Dangerous applications. Risk level of the user regarding the trust attribute is low.

S5: User is in the Company’s Premises, during the Working Hours, connected to the Company’s WiFi, does have an Outdated Antivirus and the device has installed only Trusted applications. Risk level of the user regarding the trust attribute is low.

The results of this experiment are shown in Table 5. The columns of the table represent the following: the policy and the scenario used for the experiment, the total risk value which is returned from the model, the decision that the UCON would have made based on the values of the attributes participating in the initial request and the given Full Policy and at the last column is the decision that the RA-UCON made based on the total risk value and the given RA-Policy.

<table>
<thead>
<tr>
<th>Policy</th>
<th>Situation</th>
<th>Risk Value</th>
<th>UCON Decision</th>
<th>RA-UCON Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP1/RAP1</td>
<td>S1</td>
<td>High Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP1/RAP1</td>
<td>S2</td>
<td>Medium Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP1/RAP1</td>
<td>S3</td>
<td>High Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP1/RAP1</td>
<td>S4</td>
<td>Medium Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP1/RAP1</td>
<td>S5</td>
<td>Low Risk</td>
<td>Allow</td>
<td>Deny</td>
</tr>
<tr>
<td>FP2/RAP2</td>
<td>S1</td>
<td>High Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP2/RAP2</td>
<td>S2</td>
<td>Medium Risk</td>
<td>Allow</td>
<td>Deny</td>
</tr>
<tr>
<td>FP2/RAP2</td>
<td>S3</td>
<td>High Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP2/RAP2</td>
<td>S4</td>
<td>Medium Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP2/RAP2</td>
<td>S5</td>
<td>Low Risk</td>
<td>Deny</td>
<td>Allow</td>
</tr>
<tr>
<td>FP3/RAP3</td>
<td>S1</td>
<td>High Risk</td>
<td>Allow</td>
<td>Deny</td>
</tr>
<tr>
<td>FP3/RAP3</td>
<td>S2</td>
<td>Medium Risk</td>
<td>Allow</td>
<td>Deny</td>
</tr>
<tr>
<td>FP3/RAP3</td>
<td>S3</td>
<td>High Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP3/RAP3</td>
<td>S4</td>
<td>Medium Risk</td>
<td>Allow</td>
<td>Allow</td>
</tr>
<tr>
<td>FP3/RAP3</td>
<td>S5</td>
<td>Low Risk</td>
<td>Allow</td>
<td>Allow</td>
</tr>
<tr>
<td>FP4/RAP4</td>
<td>S1</td>
<td>High Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP4/RAP4</td>
<td>S2</td>
<td>Medium Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP4/RAP4</td>
<td>S3</td>
<td>High Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP4/RAP4</td>
<td>S4</td>
<td>Medium Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP4/RAP4</td>
<td>S5</td>
<td>Low Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP5/RAP5</td>
<td>S1</td>
<td>High Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP5/RAP5</td>
<td>S2</td>
<td>Medium Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP5/RAP5</td>
<td>S3</td>
<td>High Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP5/RAP5</td>
<td>S4</td>
<td>Medium Risk</td>
<td>Deny</td>
<td>Deny</td>
</tr>
<tr>
<td>FP5/RAP5</td>
<td>S5</td>
<td>Low Risk</td>
<td>Allow</td>
<td>Allow</td>
</tr>
</tbody>
</table>

From the experiments that we run we observed that, in 84% of the cases the UCON model and the RA-UCON model made the same decision. Although, there are also some experiments in which the decision was different. This difference is expectable, considering that the request in the proposed model takes another form. In the UCON model the request includes all the attributes that will be evaluated, in order to decide whether or not the request satisfies the Full Policy. Instead, in the RA-UCON the system administrator defines the RA-Policy at the chosen granularity level. If this level is
different from the Full Policy then the considered request is the Ag-
ggregated Request, which derives from the initial one, exploiting the
attributes generated by the AHP blocks. The observed differences of
the two models in the final decision are a trade-off among the loss
of expressiveness and the improvement on the side of performance
and policy complexity. It is worth noting at this point that the final
decision the RA-UCON made can be considered reasonable, taking
into account the risk level of the attributes participating in the
given request.

More specifically, for the second (P2) and the third (P3) policy
there are deviations since the UCON model would have allowed
access in more scenarios than the proposed model did. The reason
of this difference is that these two policies take into account a small
set of the attributes, ignoring the values of the remaining, even if
those can be considered as very risky. For example, in the second
scenario regarding the second policy the user satisfies the policy
but the device is also connected to a public network which is a very
risky situation. The proposed model, considers this risk and denies
the access. Similar example is the one of scenario one regarding
the third policy. In this scenario the policy stands, although the
device has no antivirus installed and it is already infected. Again,
the risk-aware UCON treat this request as highly risky and denies
access. There are also some cases in which the UCON denies the
access and the RA-UCON allows it. One of these cases is the one
scenario five regarding the fourth policy. The reason why UCON
denies access is that the antivirus which is installed in the device
is outdated. Notwithstanding that, the risk of this request is low
since the majority of the values encapsulates a low level of risk.
Moreover, considering that the user is also trusted (his/her level of
risk is low) it is reasonable to grant the access.

5.2 Performance Analysis
For this experiment we have defined three different sets of policies,
one with 20 attributes, one with 5 attributes and a last one with
only 1 attribute. In order to evaluate the time needed for each
evaluation, a UCON implementation based on Balana, presented
in [14] has been used. For each policy an average evaluation time
has been calculated after 10 executions of the PDP. The goal of
this experiment is to show the difference in performance, when the
PDP has to evaluate a policy with a big number of attributes and a
policy with only one attribute.

Moreover, except for the difference in evaluation time, there is
also a difference in the convenience of writing a policy consisted of
many attributes and a policy consisted of only one. In Listing 1 an
example of an one attribute policy is given in order to demonstrate
this difference. The small size of policies like this one facilitate not
only the writing but also the maintenance, the modification and
the restriction of possible errors.

### Listing 1: One attribute policy

```xml
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
        PolicyId="policy2Attributes" RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:deny-unless-permit"
        Version="3.0">
    <Target>
        <Rule Effect="Permit" RuleId="rule-permit" />
    </Target>
    <Apply>
        <AttributeValue Data-Type="http://www.w3.org/2001/XMLSchema#string" medium="medium" />
    </Apply>
    <Condition>
        <On-going-condition />
    </Condition>
</Policy>
```

...
Thus, the evaluation of a Full-Policy from the PDP can introduce the Lines of Code (LoC) needed to write the policy. However, an one attribute policy can be avoided using an one attribute policy. If the policy contains a large set of attributes, can result in errors and conflicts which can be avoided using an one attribute policy. The distributiveness and heterogeneity of architectures belonging to the IoT paradigm generates several challenges for what concerns controlling the right to access generated information and controlled resources. In fact, the number of attributes being considered by the access control policies might increase exponentially, making thus the policies more and more complex. Identified the need to reduce by aggregation the number of attributes for IoT related policies, in this paper we have proposed a qualitative risk-based model for usage control, which aggregates several attributes by exploiting the distributiveness and heterogeneity of architectures belonging to the IoT paradigm.

Table 6: Performance Results

<table>
<thead>
<tr>
<th>Policy</th>
<th>20 attributes</th>
<th>5 attributes</th>
<th>1 attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Time</td>
<td>1262 ms</td>
<td>697 ms</td>
<td>557 ms</td>
</tr>
<tr>
<td>Maximum Value</td>
<td>1623 ms</td>
<td>933 ms</td>
<td>601 ms</td>
</tr>
<tr>
<td>Minimum Value</td>
<td>649 ms</td>
<td>575 ms</td>
<td>502 ms</td>
</tr>
<tr>
<td>LoC</td>
<td>456</td>
<td>140</td>
<td>56</td>
</tr>
</tbody>
</table>

Regarding the performance analysis the results show that the 1 attribute policy in comparison with the 20 attributes policy is being evaluated up to 50% faster. Also, comparing it to the 5 attributes policy we can also observe a better performance. The difference in the timing behavior is expected, considering that a policy with a large number of attributes demands the acquisition of all the corresponding values, procedure that can be turned out time consuming. Thus, the evaluation of a Full-Policy from the PDP can introduce an additional overhead to the decision making procedure. On the contrary the assessment of a RA-policy, which takes into account a smaller number of attributes, results in a faster evaluation time. Last but not least, the difference regarding the effort which a security manager has to make to write the policy, is more than obvious. The one attribute policy needs only 56 Lines of Code, while a twenty attribute policy needs 400 lines more. Consequently, the writing, maintenance, modification and correction of a Full-policy, consisted of a large set of attributes, can result in errors and conflicts which can be avoided using an one attribute policy.

6 RELATED WORK

In UCON the continuous refresh of the attributes is mandatory. Many times this is a difficult task to accomplish, due to networks delays, hardware failures etc. In [10] the authors made a first attempt with their study to introduce the notion of risk in usage control and dealt with the problem above using the Markov chains in order to model this mutability of the authorization context. Also, in [12] the authors proposed a cost-effective enforcement of UCON policies, where they try to identify and estimate the impact of all uncertainties associated with attributes acquisition. However, their proposal focuses only on the risk that the attribute acquisition encapsulates and doesn’t take into account the risk that the from the other attributes. In fact, the risk of the attribute acquisition could constitute a block in the reverse tree of the proposed model. Authors of [9] presented a framework for UCON enforcement in GRID systems, extending the classical UCON model by adding trust and risk management functionalities. In order to make a decision about allowing or denying access, they conduct a risk assessment procedure by comparing the resulting cost in case access is allowed with the cost when it is revoked. Although, their framework is designed for GRID systems and when it comes to the risk calculation they take into account only trust and reputation metrics of the participating entities. Another attempt to add the notion of risk in the UCON model was made in [11], where we meet a qualitative risk-based decision process that can be applied to usage control model in Service Oriented Architectures. In these kind of architectures, data providers need to select the data consumers which have the lowest risk level and the proposed process provides a solution to that. However, this study is specific for SOA and the risk refers to the risk level of the data consumer. Another approach is the one in [21], where the authors propose two dynamic risk-based decision methods for typical access control systems. Both of these methods are based on awarding reward and penalty points to users after evaluating the way that they used their access privileges. The risk factor is calculated based on the sensitivity of the object and the clearance level of the subject. With respect to this work, it is also a model that takes into account a partial risk that is related to the trustworthiness of the requester. In [8] AHP and fuzzy comprehensive evaluation model are combined to assess the risk of an accounting system. The authors build their model according to the three main factors of the AHP, the goal, the criteria and the alternatives. Also, in [7] the authors propose a risk assessment methodology for an R&D project using the same procedure. Both of these two studies exploit AHP in order to assess the risk, but they don’t apply their framework in access/usage control decisions.

7 CONCLUSION
the associated risk level of each value, in relation to performing a critical operation. This model is designed to simplify the process of writing risk-based policies and exploits the AHP decision process to calculate the relevance of each attribute participating in the security policy, with respect to the total risk of the request. The proposed model, by exploiting a Reverse Tree structure for attribute aggregation, can be scaled to define a hierarchy of subproblems at different levels. The proposed framework associated to this model seamlessly extends the standard Usage Control workflow, to consider the risk based attributes together with the normal ones, allowing the definition of policies of any granularity. Hence, the framework automatically manages the access and usage requests, ensuring that the attributes related to the defined policies are analyzed. In the performed experiments, the framework has demonstrated the viability of the proposed model, showing the decision equivalence and the improvement on the performance side.

As future work we plan to extend the experimental testbed, by considering more complex systems with different coexisting UCON policies. Aspects related to possible conflicts in policies, with conditions having opposite effect on risk will be considered. Furthermore, other attribute aggregation strategies will be considered, comparing also the current qualitative risk model with a quantitative one.

ACKNOWLEDGMENTS

This work has been partially funded by EU Funded project H2020 NeCS, GA #675320 and H2020 C3ISP, GA #700294 and EIT Digital Trusted Cloud and Internet of Things.

REFERENCES