A Dynamic Risk-based Access Control Architecture for Cloud Computing

Daniel Ricardo dos Santos, Carla Merkle Westphall and Carlos Becker Westphall
Networks and Management Laboratory
Federal University of Santa Catarina
Florianópolis - SC - Brazil
{danielrs,carlamw,westphal}@inf.ufsc.br

***Abstract*—Cloud computing is a distributed computing model that still faces problems. New ideas emerge to take advantage of its features and among the research challenges found in the cloud, we can highlight Identity and Access Management. The main problems of the application of access control in the cloud are the necessary flexibility and scalability to support a large number of users and resources in a dynamic and heterogeneous environment, with collaboration and information sharing needs. This paper proposes the use of risk-based dynamic access control for cloud computing. The proposal is presented as an access control model based on an extension of the XACML standard with three new components: the Risk Engine, the Risk Quantification Web Services and the Risk Policies. The risk policies present a method to describe risk metrics and their quantification, using local or remote functions. The risk policies allow users and cloud service providers to define how to handle risk-based access control for their resources, using different quantification and aggregation methods. The model reaches the access decision based on a combination of XACML decisions and risk analysis. A prototype of the model is implemented, showing it has enough expressivity to describe the models of related work. In the experimental results, the prototype takes between 2 and 6 milliseconds to reach access decisions using a risk policy. A discussion on the security aspects of the model is also presented.**

I. Introduction

Cloud computing is a model for delivering computational resources and services to users through the Internet, providing features such as easy access, elasticity and resource sharing [1].

With the development of cloud computing and a broader adoption of the model, new ideas emerge to leverage its features even further. Among these ideas, we can highlight cloud federations, which aim to make possible the sharing of resources among several clouds grouped in a federation.

Some problems still have to be faced in the scenarios of cloud computing and cloud federations, and those related to security deserve special attention [2]. The increasing number of users and resources available in the cloud, coupled with the great dynamism and heterogeneity of this environment makes it necessary to securely and efficiently manage who are these users and which resources they can access.

Identity and Access Management (IAM) is fundamental to ensure characteristics such as privacy, confidentiality and integrity of data in the cloud [3, 4]. Although access control is fundamental for cloud security [5], traditional access control models, which are still the most widely implemented in the cloud, present problems in this kind of environment.

978-1-4799-0913-1/14/$31.00

Dynamic access control models, such as those based on risk and context, were developed to deal with the problems of highly dynamic environments [6]. Also, these models are able to deal with exceptional access requests, when a normally unauthorized user must be granted access to perform a critical action. This is known as "break the glass".

The main issue solved by this kind of access control model is flexibility in accessing resources. Traditional models employ rigid and static access control policies. These policies reach their intended security goals, but are not well suited to dynamic and heterogeneous environments like the cloud, which present a constant change in the available users and resources.

This paper presents a model for dynamic risk-based access control for cloud computing. The system manages the access of users to cloud resources using the quantification and aggregation of risk metrics that are defined in risk policies, which are created by the owners of the resources. The risk-based model is built on top of an eXtensible Access Control Markup Language (XACML) architecture and, therefore, allows the use of Attribute-based Access Control (ABAC) coupled with the risk analysis. This combination provides great flexibility for access control for both the users and the Cloud Service Providers (CSP).

The rest of this paper is organized as follows: Section II describes the basic concepts of cloud computing, cloud federations, identity and access management; Section III presents the proposed model; Section IV describes the implementation and results; Section V discusses the related work; and Section VI is the conclusion.

II. Cloud Computing, Identity and Access Management

The cloud computing paradigm has been successful because of its scalability and reduced costs, but some authors claim that in order to use its full potential, a step must be taken toward cloud federations [7].

A. Cloud federations

A cloud federation comprises services from different providers aggregated in a set that supports three basic interoperability features: resource migration, resource redundancy

and the combination of complementary resources or services [8].

Several proposals and architectures for cloud federations are being discussed in the literature, but they all share the same idea of aggregating sets of clouds through the use of standard protocols, allowing them to interact and utilize resources of one another. This is also known as multi-clouds or clouds of clouds [9, 10].

The main benefits of this model are an increase in scalability and availability, as well as reduced costs, because providers can outsource their resources. It is also expected that the migration of resources improves interoperability among clouds, avoiding problems such as vendor lock-in.

There are also proposals for the creation of an Intercloud, aggregating clouds in a global scale the same way that the Internet aggregates networks [11, 12]. Some of the main cloud federation works being developed are funded by the European Union: Contrail, Reservoir and mOSAIC [13].

B. Identity and Access Management

Identity and Access Management comprises the processes related to the identification, authentication, authorization and accountability of users in computer systems. Authorization or access control is the process through which the system ensures that access requests are validated with well-defined rules [14].

Those rules are known as policies and the way that these policies are defined and managed constitutes an access control model.

In Federated Identity Management (FIM), digital identities are shared among *Users*, *Identity Providers* (IdP) and *Service Providers* (SP). A federation is an association comprised by any number of SPs and IdPs [15]. Trust is implicit in this definition [16], with every participant being expected to trust the others, in what is known as a Circle of Trust (CoT).

The main problems with the FIM approach are the need for negotiating the CoT, which can hinder dynamic collaboration [17] and the use of an extensive number of protocols and standards, which reduces interoperability. Those problems lead to a reduced scalability in practical applications [18].

At this point, it is important to make a distinction between cloud federations and identity federations. Cloud federations share resources among different CSPs, while identity federations share identity information among different domains. The trust requirements and assumptions are not the same in each case.

An access control system considers *Subjects* trying to execute *Actions* on *Resources* and is comprised of policies, which describe what is permitted in the system, and mechanisms for enforcing the policies.

Access control systems are categorized into models, and the most traditional models are Discretionary Access Control (DAC), Mandatory Access Control (MAC) and Role-based Access Control (RBAC) [19]. However, the emergence of new system architectures such as Web-based and other distributed systems led to the development of new models, among them Attribute-based Access Control (ABAC) and Usage Control (UCON).

Despite the fact that unauthorized disclosure, denial of service and data tampering are still critical, new kinds of systems, characterized by distribution, automatic reconfiguration and dynamism present new challenges for access control systems. Models such as cloud computing, pervasive and ubiquitous computing and computer grids fit those needs [20, 21].

C. Risk-based access control

Traditional access control models rely on static authorization, i.e., every access decision is pre-established, based on the policies. The idea behind dynamic access control systems is that every access request must be analyzed in its context, dynamically, taking into account not only the policies, but also contextual information such as security risk, operational need and benefit of the action for the system and the users, among others.

In real applications, unexpected situations often require the violation of security policies. This may occur because policies are incomplete or incoherent, sometimes even conflicting. The most usual examples of such needs are in medical and military applications, where the need to take actions may save lives and system stagnation may cause serious harm. The support for this kind of situation is known as “break the glass” and it is an approach for providing flexibility to policies [22].

Dynamic access control models are characterized by the use of a function that evaluates in “real time” each access request. Features that can be taken into account by this function include risk, need, benefit, trust and context. The dynamic nature of access control is captured in these models because access decisions may vary according to contextual information evaluated at the time of the request.

Risk is the potential damage that can arise from a process and is usually represented by the probability of occurrence of an undesired event multiplied by its impact [23]. Risk metrics are a way to quantify assets, threats and vulnerabilities of a system. Also, risk is different from uncertainty, because risk can be measured and managed [24].

Risk-based access control systems perform a risk analysis on access requests to reach an access decision. This analysis can be qualitative or quantitative, automatically attributing a numeric value to risk.

Risk quantification is easier in situations where there is a history of events and impact can be easily measured. Also, other components besides probability and impact may be taken into account. These are the reasons why there are different risk-based access control models.

The Risk-adaptive Access Control (RAdAC) model, developed by the NSA, is a pioneer and best suited for military applications [6].

Some challenges to achieving risk-based access control are: calculating security risk in real time; determining operational need; quantifying trust level; using heuristics to reach access decisions; and allowing access revocation at any time [25].

Some approaches to risk quantification use fuzzy logic [26, 27], others employ machine learning [28] and still others use probabilistic inference, decision theory etc.

D. Architectures for access control systems

The main reference architecture for access control is presented in RFC2904 [29]. It defines four components for an access control system: the Policy Retrieval Point (PRP), where policies are stored and retrieved; the Policy Information Point (PIP), where information useful for access decisions are retrieved; the Policy Decision Point (PDP), where policies are evaluated and access decisions are achieved; and the Policy Enforcement Point (PEP), which protects sensitive resources and forwards access requests to the PDP.

XACML is a standard for access policies, requests and responses, as well as a reference architecture for access control systems [30]. It is based on RFC2904, but renames the PRP to Policy Administration Point (PAP).

III. A MODEL FOR DYNAMIC RISK-BASED ACCESS CONTROL IN CLOUD COMPUTING

The access control model proposed in this work is risk-based and employs the notion of quantifying risk metrics and aggregating them. It also presents the idea of risk policies, which allow CSPs and resource owners to define their own metrics, allowing greater flexibility to access control.

Figure 1 shows an overview of the model, which is an XACML extension, and its components. The following new components were added to the XACML core components:

- **Risk engine** - Called by the PDP to process risk-based access control. Responsible for analyzing and processing the risk policies associated to a resource and invoking the risk quantification and aggregation methods described in each one. It is different in each CSP, because it implements locally the quantification methods which are available in the CSP. If users want to use other methods, they must provide an implementation of these methods in the form of a web service, whose URL is in the risk policy;
- **Risk Quantification Web Services** - Responsible for quantifying the risk in every access request. It is possibly implemented by users. Each web service is responsible for receiving an access request forwarded by the risk engine and returning a numeric value that represents the quantification of the risk metric; and
- **Risk policies** - Define how risk-based access control must be evaluated for each resource.

In the proposed extension, when an access request is received, the PDP may perform two access control decisions in parallel. On the one hand, the PDP performs the ABAC decision, based on the XACML policies related to that resource. On the other hand, the PDP and the risk engine perform the risk-based access decision, according to the risk policies.

The possible results of evaluating a risk policy are the same as those of an XACML evaluation: PERMIT, DENY, NOTAPPLICABLE and INDETERMINATE.

After evaluating all of the policies, the PDP has two access decisions: one based on the XACML policies and one based on the risk policies. The decisions may be incompatible and must be combined to achieve a single final result. To that end, four policy combination methods are defined: **Deny overrides** - the result is DENY if any of the evaluations is DENY; **Permit overrides** - the result is PERMIT if any of the evaluations is PERMIT; **ABAC precedence** - the final result is the same of the XACML result; and **Risk precedence** - the final result is the same of the risk-based result.

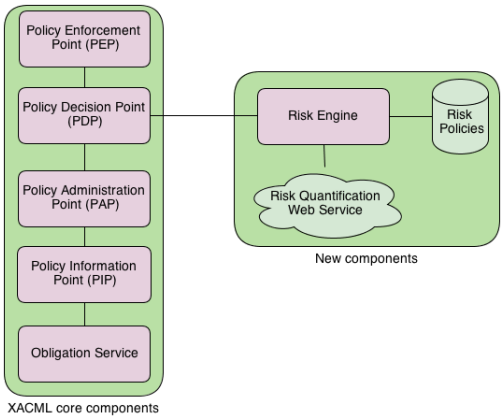

Fig. 1. Overview of the access control components

A. Risk policies

A risk policy is an XML file which describes to the CSP how the risk-based access control must be performed for a determined resource. This file is created by the owner of the resource and stored in the CSP. Each policy contains an identification of the related resource, an identification of the owner of the resource, a series of risk metrics with their descriptions and quantification methods, a risk aggregation method and an acceptable risk threshold.

Quantification methods are the functions used to give a numeric value to a risk metric, based on the access request. An aggregation method is a function which receives the risk values calculated for each metric and aggregates them in a single value. Two kinds of quantification methods are allowed: local or external. Local methods invoke functions defined in the risk engine itself, while external methods invoke web services.

It is important to highlight that the owner of a resource can always opt if the resource can be accessed through risk or not, in order to maintain the flexibility of the proposal. The CSP can also opt if it accepts its resource to be accessed this way. If the CSP agrees to risk-based access control, but the user is against it, the user decision prevails.

Besides the risk policies associated to each resource, the CSP must make available a basic risk policy. The basic policy is also an XML file, but defined by the CSP, which contains the minimum risk metrics that the system demands, as well as a minimum risk threshold.

The basic policies of each CSP are evaluated in every access request, before the specific policies of each resource. If the basic policy is violated, the risk-based access decision is immediately a DENY. In this case, the specific policies of a resource are not even processed. Therefore, the basic policies are important to maintain the minimum security requirements of a CSP, at the same time allowing flexibility in access control.

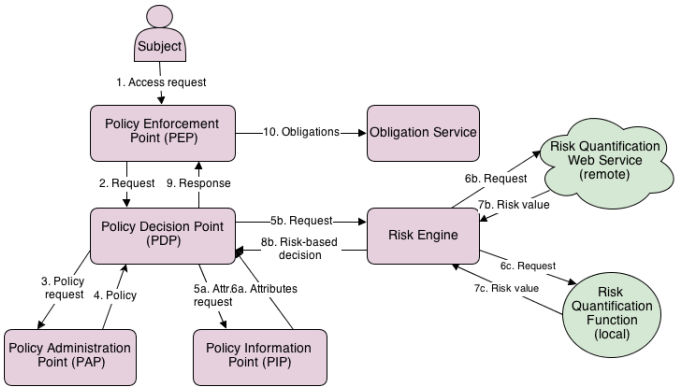

Fig. 2. Decision process step by step

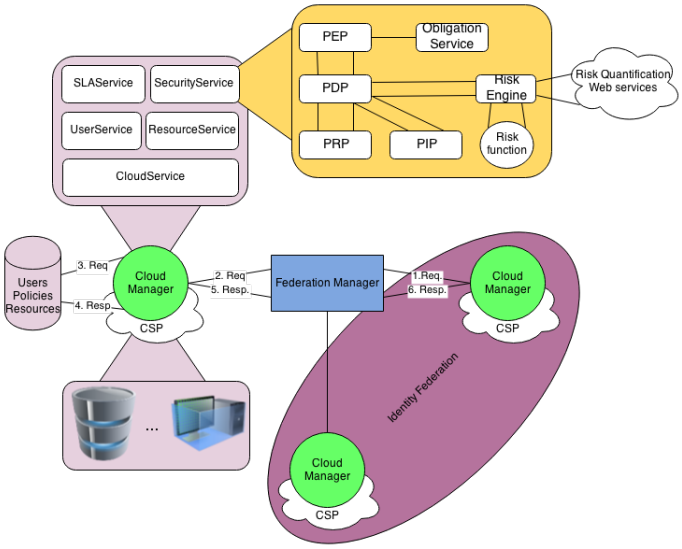

Fig. 3. Access control model inserted in a cloud federation

B. Decision process

Figure 2 presents the proposed access control decision process step by step. Step 1 is the issuing of an access request by the subject to a cloud resource. The PEP receives this request and forwards it to the PDP (step 2). The PDP requests to the PAP the XACML and risk policies associated to the resource (step 3) and gets them as response (step 4).

At this point, both access decisions take place in parallel. For the XACML decision, the PDP requests to the PIP the attributes informed in the policy (step 5a), get them as response (step 6a) and the access request is then evaluated in the traditional way by the PDP.

For the risk-based decision, the PDP first verifies if the resource may be evaluated this way. This permission must be given by the CSP and by the owner of the resource and is represented by the existence of risk policies associated to the resource. If there are no associated risk policies, the access decision is NOTAPPLICABLE.

If there are associated policies, the PDP forwards the access request to the risk engine (step 5b), which firstly analyzes the basic risk policy to which the CSP is subjected. If the evaluation of the basic policy returns PERMIT, the risk engine analyzes the risk policies and performs the quantifications according to the specifications (steps 6 and 7). If the risk quantification is done locally, a function in the risk engine itself is executed (steps 6c and 7c) and if it is done externally, a web service is invoked for the quantification (steps 6b and 7b).

Risk metrics are aggregated in a single value and the risk engine returns a decision to the PDP (step 8b). The PDP, having received the XACML and risk decisions, applies one of the policy combination rules, which is previously defined by the CSP, and decides to grant the access or not, sending the response to the PEP (step 9). The PEP is then responsible for analyzing and applying obligations (step 10).

C. Cloud Federations

One of the greatest challenges in establishing and maintaining a cloud federation is IAM [31]. For a CSP to trust the identity information of users from another CSP, they both must share some agreement of trust, thus this process is usually mediated by an identity federation.

However, as stated in section II, the identity federation approach presents two big problems in real scenarios: trust agreements and interoperability. We propose to use our risk-based access control model to allow the use of cloud federations without the need for identity federations. This part of the model is presented in more details in dos Santos et al. [32].

To solve the issue of trust agreements in identity federations, we take advantage of the fact that the establishment of a cloud federation represents a level of trust among providers, however lesser than in an identity federation. The participating clouds may not trust identity information shared among them and this need for trust can be filled by a risk metric in access control.

The issue of interoperability among federations cannot be fully solved by using risk-based access control, but it is alleviated. Interoperability has to happen in two levels: in the message format level and in the attribute level. In the message format level, it is still necessary that entities communicate using a standard protocol, such as Security Assertion Markup Language (SAML). Nevertheless, in the attribute level, even if the two parties do not agree on which identity attributes to use, it is still possible to use risk-based access control. In this case the access request would be treated as an exceptional access request as those described in section II, i.e., an access request by a subject that would usually have a permission denied.

To describe the use of access control in the cloud federation, we first define the architecture for the cloud federation to be used. The goal of the architecture is to ease collaboration among cloud providers. The main idea is that several clouds, public or private, can aggregate and share their resources, allowing users to instantiate their resources in any point of the federation. The federation architecture with the access control model is presented in Figure 3.

The main components of the federation are: *CloudManager* - connects a CSP to the federation and contains several

services, inside of which the access control is performed; and *FederationManager* - connects the *CloudManagers*, acting as a list of available *CloudManagers* and managing message exchange among them. As shown in Figure 3, some of the clouds may form identity federations among themselves, while others choose not to.

Users see two kinds of cloud in this scenario: a home cloud, representing their original CSP and foreign clouds, representing the other CSPs. Users may instantiate and access resources in both kinds of clouds, but each action has its particularities.

When instantiating a resource, users may choose if they wish to do so on their home cloud or on a foreign cloud. If the resource is instantiated on the home cloud, it can be private to users of that cloud or public, accessible to users of any cloud. If the resource is instantiated on a foreign cloud, it must be public. At the instantiation, users must create a XACML policy related to the resource and, optionally, a risk policy. The creation of the risk policy signals the intention of using risk-based access control for that resource. If both policies are created the user must also choose a combination rule.

When users try to access a resource instantiated in their home cloud, the request is handled by the CSP itself, with no interference from the federation components. When users try to access a resource in a foreign cloud, this request goes from one *CloudManager* to another, where it is handled by the risk-based access control model.

D. Considerations about the proposal

The use of risk-based access control for cloud computing allows the possibility of a great flexibilization in accessing resources and information. This use, however, may also bring problems, since risk analysis may be a very subjective process. Therefore, the choice of using risk-based access control must be made firstly by the CSP and then by the users. Also, supporting obligations, embedded in XACML3.0, is important for monitoring and accountability.

Using XACML and risk decisions in parallel and combining them creates an array of possibilities for access control. Dynamic risk-based access control can have a greater or smaller influence in the final decision. If the system uses only ABAC, risk has no influence in the decision; if it uses only risk, ABAC has no influence; and between these extremes there are possibilities for giving priorities to one or the other or requiring both to reach the same decision.

The proposal of risk policies allows the use of diverse risk quantification methods and also allows users to define their own methods. The proposal ensures the fulfillment of minimum security requirements through the use of basic policies.

Another important characteristic of the model is the possibility of distributing and replicating policy points, to avoid single points of failure. This is inherited from XACML and kept in the extension, since the new components may also be distributed and replicated.

The main limitations of the proposal are the overhead caused by processing risk policies and the performance degradation when using web services. The performance of the proposal is analyzed in Section IV.

Issues such as secure protocols for message exchange and authentication are not discussed in the proposal because the model considers authentication to be secure, in order to focus on authorization.

The use of risk-based access control in cloud federations decreases the need for using identity federations in these environments. This happens because the establishment of a cloud federation already requires a level of trust among members and also because it becomes possible to use authentication provided by each CSP separately.

The possibility of not using identity federations has the advantages of avoiding scalability and interoperability issues caused by FIM. However, the level of trust among individual providers is diminished, which could lead to restricted access and a greater need for auditing and accountability.

In the end, the choice between identity federations or risk-based access control is a choice between a greater level of trust or greater interoperability and scalability.

IV. Experimental results

The proposed model was implemented in three stages: the access control model; the cloud federation; and the risk quantification and aggregation methods. All the implementation was done using the Python language.

The access control system uses the ndg-xacml module to handle XACML requests and the web.py framework for the web services. The implementation of the federation was simplified to highlight the access control aspects, so features such as resource migration were not implemented. The federation uses the ZeroMQ framework for message exchange; the peewee framework and the MySQL database for data persistence; and the OpenNebula cloud management software.

To test the implemented system and show the expressivity of the model, two methods presented in related work were implemented. The chosen methods were: Britton and Brown [33] and Sharma et al. [34], which uses three risk metrics: impact on Confidentiality; impact on Availability; and impact on Integrity. The paper [34] presents a table relating actions and resources showing predefined risk impacts for each metric. The risk value of each metric is aggregated according to a formula to obtain a final result. The risk policy representing the model is as follows:

Listing 1. Risk policy for Sharma et al. [34]

```
<rp:risk-policy version="1.0" xmlns:rp="http://inf.
    ufsc.br/~danielrs">
    <rp:resource id="1"/><rp:user id="1"/>
    <rp:metric-set name="sharma2012">
        <rp:metric>
            <rp:name>Confidentiality</rp:name>
            <rp:quantification>https://
                localhost:8443/quantify-conf</
                rp:quantification>
        </rp:metric>
        <rp:metric>
            <rp:name>Availability</rp:name>
            <rp:quantification>https://
                localhost:8443/quantify-avail</
                rp:quantification>
```

```
        </rp:metric>
        <rp:metric>
            <rp:name>Integrity</rp:name>
            <rp:quantification>https://
                localhost:8443/quantify-int</
                rp:quantification>
        </rp:metric>
    </rp:metric-set>
    <rp:aggregation-engine>https://localhost:8443/
        aggregate</rp:aggregation-engine>
    <rp:risk-threshold>1.5</rp:risk-threshold>
</rp:risk-policy>
```

A. Example of use

To illustrate the operation of the implementation we describe an example of use. In this example, we consider only one CSP that stores and instantiates the resources of its users.

Suppose that Alice instantiates a virtual machine (VM) in this CSP and decides that it accepts risk-based access control, which is supported by the CSP. She then defines an XACML policy, a risk policy and a combination rule for this VM. In the XACML policy, Alice defines two types of access: (i) she and users who belong to her group of friends can view the machine; and (ii) only she can edit or delete the machine. All of the other actions, for all of the other users are forbidden. As risk policy, Alice uses the implementation of Sharma et al. [34], which consider metrics for Confidentiality, Integrity and Availability.

Suppose that there are two more users in this CSP: Bob, who is in the group of Alice's friends and Charlie, who is not. Considering the XACML access control, when Alice tries to access the machine for any action, her access is granted; when Bob tries to access the machine, he has the access granted for viewing, but not for editing or deleting; Charlie has the access denied for any action. Considering risk, the result is the same for any user because in the implementation the past risk score was fixed in 1 for every user.

Let us explore an access decision for Charlie's request to view Alice's VM. The request comes to the PEP and is sent to the PDP. At this moment the XACML PDP and the risk engine are called. In this case, the XACML decision would be DENY. The risk decision is based on the action chosen (viewing) and the fact that the resource is sensitive. Thus, the impact results would be: 0 for availability (a), 0 for integrity (i) and 1 for confidentiality (c) [34]. Since we have no history to base our calculations, let us consider the probability of occurrence of every result as 0.33 and the past risk score of every user as 1. The aggregated risk would then be:

((a * p1) + (i * p2) + (c * p3) + pastScore) = ((0 * 0.33) + (0 * 0.33) + (1 * 0.33) + 1) = 1.33

Since the risk score (1.33) is lower than the threshold defined in the policy (1.5), the risk decision is PERMIT. The final result depends on the combination rule being used. If the rule is Permit Overrides or Risk Precedence, the result is PERMIT, otherwise it is DENY.

B. Experiments

To test the implementation, we used VMs instantiated in Amazon EC2. These machines have 1.7 GB of RAM, 160GB of storage and a CPU that corresponds to a 1.2GHz Xeon. Three sets of experiments were performed. The first set is a comparison among different access control policies. The second set is an evaluation of the number of metrics in a given policy and the third, an evaluation of the influence of local and external metrics in the same policy. All of the times are in milliseconds (ms).

The results of the first set of experiments are shown in Table I, which presents the time spent to reach an access decision using three different policies: (i) only XACML; (ii) XACML + the policy of [34]; and (iii) XACML + [33]. All of the quantification and aggregation functions are implemented locally. The increasing time is due to an increasing number of metrics.

TABLE I. PERFORMANCE OF RISK POLICIES

Policy	min. (ms)	max. (ms)	avg (ms)
XACML	0.925	4.278	1.040
XACML+[34]	1.986	11.973	2.436
XACML+[33]	4.395	14.234	5.352

In the second set of experiments, we used a risk policy with a varying number of metrics, all quantified locally. All of the metrics just returned random values, so we could get a performance result based only on the number of metrics and not on the complexity of each metric. Table II shows the results of this set of experiments.

TABLE II. PERFORMANCE WITH A VARYING NUMBER OF METRICS

Number of metrics	min. (ms)	max. (ms)	avg (ms)
1	1.832	12.130	2.243
10	2.612	12.876	3.171
100	10.922	60.442	14.030
1000	96.041	175.245	121.383
10000	1168.511	1517.364	1361.025

Increasing the number of local metrics impacts the performance of the system, however, this impact can be tolerated even with a huge number of metrics (10000). It is important to notice that the impact on performance is due more to the processing of the XML file containing the policy than to the processing of the metrics.

In the third set of experiments, we used a risk policy containing 10 policies which, as before, return random risk values. In this set, four kinds of policies were defined. Case A represents 10 requests handled only by local XACML; case B represents 10 local risk quantification metrics; case C represents 5 local and 5 remote metrics (web services); and case D represents a risk policy with 10 remote metrics. In every case the aggregation rule is local. Table 3 shows the results obtained in each case.

TABLE III. PERFORMANCE WITH LOCAL AND EXTERNAL METRICS

Case	min. (ms)	max. (ms)	avg (ms)
A	1.057	9.372	1.46
B	1.824	15.564	4.574
C	1556.182	2813.56	1726.71
D	3247.563	10350.5	4220.6

It is easy to notice that the use of web services heavily impacts performance and that the use of 10 remote metrics is already impracticable for an access control system. Finally,

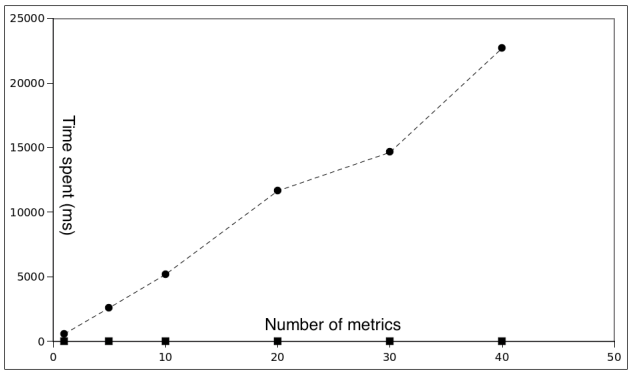

Fig. 4. Time spent to reach an access decision

Figure 4 shows the growth in time spent to reach an access decision as the number of metrics increase. The X axis is the number of metrics and the Y axis is the time spent in milliseconds. The solid line with square markers is for local metrics and the dashed line with circular markers is for remote metrics.

C. Discussion

Measuring security is not easy and for access control systems it usually involves the definition of a set of possible states and the proof that in no configuration of states there is leaking of permissions [35]. Since there is no formal modeling for risk-based access control, it is not possible to prove its correctness and, therefore, its security. This is why there are no experiments related to the security of the system presented in this paper.

Hu et al. [35] recommend assessing an access control system based on: administrative capacity and costs, policy coverage, extensibility and performance.

The proposed model is equivalent to XACML in terms of administrative capacity and costs, only adding the need to manage risk policies. The coverage of policies and extensibility of the system can be shown through the implementation of models presented in related work. The performance of the system was evaluated quantitatively and we can draw some conclusions from the experiments. It can be seen that using the system with local metrics presents a satisfactory performance and despite a performance decrease with a bigger number of metrics, this is expected and the system does not become inefficient. The use of external metrics, however, heavily impacts the system, because of the time spent in HTTP communication.

V. Related Work

Fall et al. [21] focus on the authorization problems created by multi-tenancy in the cloud. The authors argue that traditional access control models are static and not well suited to the cloud, while risk-based models are dynamic and naturally adapted to this environment. The authors propose using the NSA RAdAC model and identify some risk situations for the cloud. The paper introduces the concept of risk-based access control for cloud computing, but shows no implementation.

Arias-Cabarcos et al. [36] describe current issues in FIM on cloud and propose using risk analysis to allow dynamic federations. The authors propose using risk metrics to quantify and aggregate risk and present a taxonomy of risk metrics considering pre-federation and post-federation stages. There is an example of use and the method is detailed. The proposal, however, considers a fixed set of metrics, not allowing users or providers to define their metrics.

Sharma et al. [34] show a risk-based access control model for cloud e-health. According to the authors, RBAC does not take into account uncertainty and risk, thus being unsuited for the cloud. The paper presents a prototype implementation considering three metrics: confidentiality, availability and integrity. In the model, every task to be accomplished in the system is sent to the cloud, where a risk score is attributed to it. The model is implemented on top of RBAC, so it uses role delegation along with the risk analysis.

Britton and Brown [33] present a quantification method for the NSA RAdAC model. In their proposed model, 27 metrics are divided in 6 categories, evaluated for every access request and aggregated to achieve a measure of the total security risk. Their risk definition considers both probability and impact as high, medium or low. They employ a triangular probability distribution and a Monte Carlo simulation to find the probability of each event, which is then multiplied by a weight attributed by experts to each metric. Since it is a method for military applications, some metrics are not suitable for a general cloud application.

Several works describe authentication and authorization in different cloud federation models [37, 38, 39, 31].

This work is an improvement on our previous work [32], which focused on the application of a similar model to cloud federations. In the present work, the model has been revised and we present new experiments and a greater analysis and discussion of the model.

VI. Conclusion

The development of access control systems for cloud computing is of great importance, because these systems are fundamental to enable the security of these environments.

Traditional access control models, currently implemented in most cloud solutions are not enough to ensure the security of these environments when it is necessary to have a greater flexibility to enable efficient information sharing in critical situations.

Risk-based access control models are an alternative and, despite the fact that there are proposals in the literature for its use in the cloud, they are very specific to a given situation, disallowing its application in a more general context and there is no reference architecture that allows its extension.

This paper presented a dynamic risk-based access control architecture for cloud computing, with an application to cloud federations. The architecture is built as an XACML extension, adding flexibility for resource and information sharing in a dynamic environment such as the cloud, while keeping the distribution and scalability features. The architecture is based on the use of risk policies, which describe the risk metrics considered most important by users and providers.

A prototype of the architecture was implemented, using the risk metrics and quantification of Sharma et al. [34]. The implementation showed satisfactory performance, except when considering the use of many remote quantification rules. In comparison to the related work, this is the only one that presents the idea of risk policies and also the only one to consider aspects of risk-based access control in a cloud federation.

As future work, there are many possibilities: integrating the access control model in a mature cloud federation project; implementing other risk quantification methods to evaluate the need for new components; improving the performance of external metrics using caches or concurrent requests; and developing a reference set of risk metrics for the cloud.

REFERENCES

[1] P. Mell and T. Grance. (2011) The nist definition of cloud computing. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[2] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *Internet Computing, IEEE*, vol. 16, no. 1, pp. 69 –73, jan.-feb. 2012.

[3] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," *Security & Privacy, IEEE*, vol. 9, no. 2, pp. 50 –57, march-april 2011.

[4] E. Bertino and K. Takahashi, *Identity Management: Concepts, Technologies, and Systems*. Artech House, 2011.

[5] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583 – 592, 2012.

[6] JASON Program Office, "Horizontal integration: Broader access models for realizing information dominance," MITRE Corporation, Tech. Rep., 12 2004.

[7] P. Harsh, Y. Jegou, R. G. Cascella, and C. Morin, "Contrail virtual execution platform challenges in being part of a cloud federation," in *Proceedings of the 4th European conference on Towards a service-based internet*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 50–61.

[8] T. Kurze, M. Klems, D. Bermbach, A. Lenk, S. Tai, and M. Kunze, "Cloud federation," in *The Second International Conference on Cloud Computing, GRIDs, and Virtualization*, september 2011, pp. 32–38.

[9] M. AlZain, E. Pardede, B. Soh, and J. Thom, "Cloud computing security: From single to multi-clouds," in *45th Hawaii International Conference on System Science*, jan. 2012, pp. 5490 –5499.

[10] M. Vukolić, "The byzantine empire in the intercloud," *SIGACT News*, vol. 41, no. 3, pp. 105–111, Sep. 2010.

[11] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, and M. Morrow, "Blueprint for the intercloud - protocols and formats for cloud computing interoperability," in *4th International Conference on Internet and Web Applications and Services*, may 2009, pp. 328 –336.

[12] R. Buyya, R. Ranjan, and R. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Algorithms and Architectures for Parallel Processing*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6081, pp. 13–31.

[13] N. Grozev and R. Buyya, "Inter-cloud architectures and application brokering: taxonomy and survey," *Software: Practice and Experience*, 2012.

[14] M. Benantar, *Access Control Systems: Security, Identity Management and Trust Models*. Springer, 2006.

[15] ITU-T, "Baseline capabilities for enhanced global identity management and interoperability," 2009. [Online]. Available: http://www.itu.int/rec/http://www.itu.int/rec/T-REC-X.1250-200909-I

[16] D. W. Chadwick, "Federated identity management," in *Foundations of Security Analysis and Design V*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5705, pp. 96–120.

[17] P. Arias Cabarcos, F. Almenárez Mendoza, A. Marín-López, and D. Díaz-Sánchez, "Enabling saml for dynamic identity federation management," in *Wireless and Mobile Networking*, ser. IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, 2009, vol. 308, pp. 173–184.

[18] K. Lampropoulos and S. Denazis, "Identity management directions in future internet," *Communications Magazine, IEEE*, vol. 49, no. 12, pp. 74–83, december 2012.

[19] P. Samarati and S. de Vimercati, "Access control: Policies, models, and mechanisms," in *Foundations of Security Analysis and Design*, ser. Lecture Notes in Computer Science, 2001, vol. 2171, pp. 137–196.

[20] G. Zhang and M. Parashar, "Dynamic context-aware access control for grid applications," in *Fourth International Workshop on Grid Computing*, nov. 2003, pp. 101 – 108.

[21] D. Fall, G. Blanc, T. Okuda, Y. Kadobayashi, and S. Yamaguchi, "Toward Quantified Risk-Adaptive Access Control for Multi-tenant Cloud Computing," in *Proceedings of the 6th Joint Workshop on Information Security*, October 2011.

[22] A. D. Brucker and H. Petritsch, "Extending access control models with break-glass," in *Proceedings of the 14th ACM symposium on Access control models and technologies*. New York, NY, USA: ACM, 2009, pp. 197–206.

[23] N. N. Diep, S. Lee, Y.-K. Lee, and H. Lee, "Contextual risk-based access control," in *Security and Management*, 2007, pp. 406–412.

[24] G. Peterson, "Introduction to identity management risk metrics," *Security & Privacy, IEEE*, vol. 4, no. 4, pp. 88–91, 2006.

[25] B. Farroha and D. Farroha, "Challenges of operationalizing dynamic system access control: Transitioning from abac to radac," in *IEEE International Systems Conference*, march 2012, pp. 1–7.

[26] P. C. Cheng, P. Rohatgi, C. Keser, P. Karger, G. Wagner, and A. Reninger, "Fuzzy multi-level security: An experiment on quantified risk-adaptive access control," in *IEEE Symposium on Security and Privacy*, 2007, pp. 222–230.

[27] Q. Ni, E. Bertino, and J. Lobo, "Risk-based access control systems built on fuzzy inferences," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. New York, NY, USA: ACM, 2010, pp. 250–260.

[28] I. Molloy, L. Dickens, C. Morisset, P.-C. Cheng, J. Lobo, and A. Russo, "Risk-based security decisions under uncertainty," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*. New York, NY, USA: ACM, 2012, pp. 157–168.

[29] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence, "AAA Authorization Framework," RFC 2904 (Informational), Internet Engineering Task Force, Aug. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2904.txt

[30] OASIS. (2003) A brief introduction to xacml. [Online]. Available: https://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html

[31] D. N. Sriram, "Federated identitiy management in intercloud," Master's thesis, Der Technishcen Universitat Munchen, january 2013.

[32] D. R. dos Santos, C. M. Westphall, and C. B. Westphall, "Risk-based dynamic access control for a highly scalable cloud federation," in *7th International Conference on Emerging Security Information Systems and Technologies*, 2013, pp. 8–13.

[33] D. Britton and I. Brown, *A security risk measurement for the RAdAC model*, 2007.

[34] M. Sharma, Y. Bai, S. Chung, and L. Dai, "Using risk in access control for cloud-assisted ehealth," in *IEEE 14th International Conference on High Performance Computing and Communication, 2012*, 2012, pp. 1047–1052.

[35] V. Hu, D. Ferraiolo, and D. R. Kuhn, "Assessment of Access Control Systems, Interagency Report 7316," National Institute of Standards and Technology, Tech. Rep., 2006.

[36] P. Arias-Cabarcos, F. Almenárez-Mendoza, A. Marín-López, D. Díaz-Sánchez, and R. Sánchez-Guerrero, "A metric-based approach to assess risk for "on cloud" federated identity management," *Journal of Network and Systems Management*, vol. 20, pp. 513–533, 2012.

[37] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "Security and cloud computing: Intercloud identity management infrastructure," in *19th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises*, june 2010, pp. 263 –265.

[38] D. Bernstein and D. Vij, "Intercloud directory and exchange protocol detail using xmpp and rdf," in *6th World Congress on Services*, july 2010, pp. 431 –438.

[39] M. Coppola, P. Dazzi, A. Lazouski, F. Martinelli, P. Mori, J. Jensen, I. Johnson, and P. Kershaw, "The contrail approach to cloud federations," in *Proceedings of the International Symposium on Grids and Clouds*, 2012.