2015

RESEARCH/TECHNOLOGY/MANAGEMENT (RTM) Journal

Research article

Toward PMI-Based access control and management framework for infrastructure as a service (IaaS) providers

Gholamreza Kiani[1], Mohammad Ali Doostari[2]

1-MA student in Information Technology, University of Shahed, Tehran,Iran

gh.kiani@shahed.ac.ir

2-Assistant Professor in the Department of Computer, University of Shahed, Tehran,Iran

doostari@shahed.ac.ir (Corresponding Author)

Abstract Cloud computing service providers are the main players of the internet. One of the most important challenges that public clouds are facing is structural management of costumers. Presented solution in this paper is based on PMI Certificates and Privilege Management regarding these Certificates. Meanwhile it has been designed by observing service providers for virtual machines and increasing security and efficiency of the particular cloud computing Scope. The Objective is using of advantages of X.509 Certificates in PMI infrastructure in implementing a framework that is extendable for service providers with distributed environments and in the same time is able to support one of the main requirements of these environments i.e. moving customers safely and securely in various places. Results of this research can easily be used among cloud computing service providers that are mostly short ages. Implementation of this framework has been done and it is publicly available for research purposes.

Keywords: Cloud Computing, Iaas providers, access control, XCloud framework, X.509 PMI

1. Introduction

The research the report of which has been presented in this paper has been planned with the motivation to increase performance in efficient management of clients and service provider components in a cloud-computing services service provider. Beyond this objective, the services this framework can offer are also applicable to other distributed environments that require security and clients-management services. The present status of these service providers is based on a factor: use of static user accounts, general classifications of clients into large work groups, or utilization of security policies on the basis of applications. As the framework designed and implemented in this research, XCloud supports privilege management based on Privilege Management Infrastructure (PMI) standards in order to improve the application of security policies. Dynamic generation of user accounts and granting privileges to these accounts, and management of issuance and revocation of these accounts in the form of X.509 digital certificates are some of this framework's objectives. The current problems in this area are due to the ways in which clients are authorized and given access, and in this paper, we attempt to offer a method eventually solving these problems[1]. Regarding static user accounts, we face problems such as limited scalability and development of the organization's security boundaries, which naturally increases security risks. In the second case, general classifications of users in general environments create a fully static environment—which can hardly be altered. For example, with the various work groups and different resources, application of security policies onto each resource for each group would be very complicated. In the third case, lack of coordination among applications providing systems with security causes malfunction in multi-agent environments[2].

2. Background and related work

2-1. Cloud computing

To understand this framework, it seems necessary to get familiar with the existing structure types for cloud computing and to examine the security requirements of these structures. The collection of software or hardware resources offered to meet users' needs on the Internet is called cloud computing. Software resources like applications and hardware resources like data centers and their components can be considered. Cloud computing is a class of computational solutions in which the technology or service allows users upon requirement to access computational resources based on requests[3]. This request-based authorization has made it a complicated task for service providers to manage clients. Any service known today as cloud computing certainly fits one of the following three classes:

- Software as a Service (SaaS) is used to refer to specific software made accessible by its developers to the public on the Internet usually via an Internet explorer. For instance, e-mail services offered by famous companies on the Internet replacing old e-mail software on users' systems.
- Platform as a Service (PaaS) is an environment that provides whatever a web developer needs to create an application. This environment reduces costs and task complexity for software developers.
- Infrastructure as a Service (IaaS) allows companies active in the field of computer and information technology to produce and present their target products with minimum costs possible by taking control of different resources like servers, network tools, and storage tools. Since these services are particular to companies and organizations, they are not used for free.

XCloud has been designed as a framework for environments that offer IaaS services. In this paper, what is meant by an infrastructure is a complete virtual machine capable of offering all functions of an individual computer system. For this purpose and to implement the virtualization, use has been made of the solutions offered by the VMWare Company, one of the pioneers of the virtualization industry.

2-2. Privilege Management Infrastructure

Privilege Management Infrastructure (PMI) is used to refer to the user authorization process based on the ITU-T proposal in accordance with the X.509 standard[4]. The version presented in 2012 for X.509 certificates states the specifications of a Privilege Management Infrastructure based on X.509 ACs[1]. It can simply be taken as if this infrastructure authorizes an identity authenticated by a Public Key Infrastructure (PKI). Standard methods used today for authorization have little diversity. One of the best-known among these methods widely

[1] X.509 **A**ttribute **C**ertificates

used in many applications is XACML[2], developed and updated by OASIS. This paper is not aimed at describing the structure of this language. Its operation, however, is explained in order to identify the advantages and disadvantages of common authorization methods and to introduce the major entities used in any authentication system. In every system of authentication and authorization, 2 basic entities play roles:

1- Policy Decision Point (PDP)

2- Policy Enforcement Point (PEP)

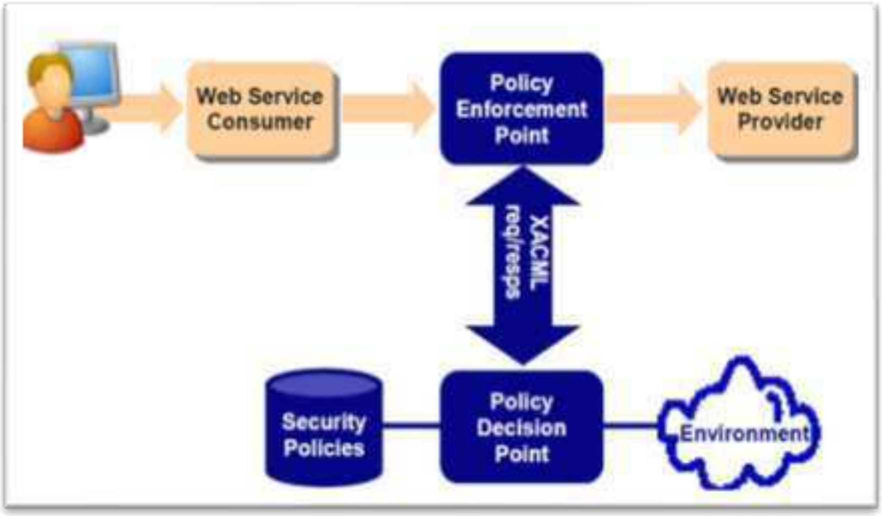


Figure 1- An Access Control System Key Elements

The language structure of XACML, as a language standardized based on XML, has been developed for communication between these two entities and is made use of in security systems[5].As displayed in the figure, the language uses the "request-response" method for communication. XACML can also be used to express the security policies in PDP. These policies are defined based on environmental variables like a computer's instant load or people's roles and other features. From this introduction, it can be concluded that to express people's roles and other particular features, a method is needed for expression and transfer of these features. The method used in this paper to express these features in the form of a secure protocol is to use X.509 ACs. Methods commonly used in authorization systems have been developed based on the idea of membership in groups. These methods are inherently defective[6]. The first reason stated to explain this inherent defect is that in this method of authorization, the possibility to assign "feature 11"s to all members creates a kind of complexity that makes it difficult to access some advantages like privileges delegation, which is an obvious characteristic of a modern authorization system. On the other hand, this method also has security shortcomings. That is, a mechanism for digitally signing these memberships and features of groups has not been presented yet. With all these explanations and structural shortcomings in current authorization systems, our need for standard X.509 Attribute Certificates is clearly felt[4].

The X.509 standard has 2 different types of certificates: Public Key Certificates (PKCs) and Attribute Certificates (ACs). It is important in this paper to identify the differences between

[2] eXtended Access Control Markup Language

these 2 types of certificates. Public Key Certificates are used for authentication, and allow their holders to introduce themselves, and the authentication is guaranteed using the Security Public Key[7]. On the other hand, Attribute Certificates are used for authorization of their holders. These certificates can be valid for different durations, and AC usually has shorter validity. Besides general fields present in all certificates, any number of desired attributes can be defined in ACs. In this paper, this type of certificates is considered as the client's document of identification. In this certificate, in addition to the client's general specifications, we define attributes where the specifications of the virtual machine delivered to the client have been mentioned. These specifications and how they are presented in this certificate will be explained later in the paper.

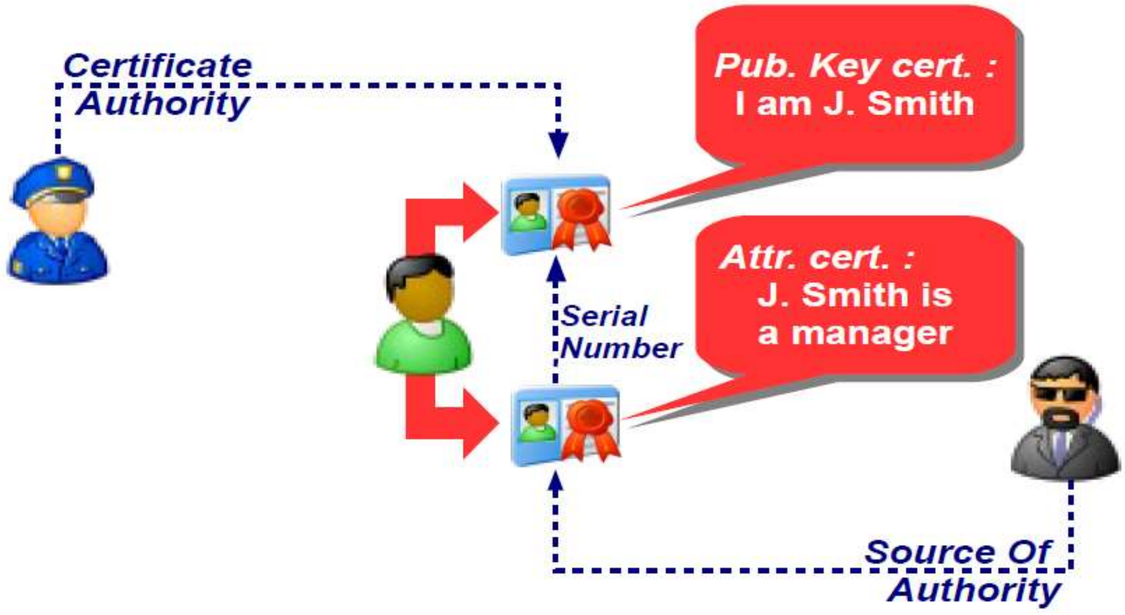

Certificate Authority
Pub. Key cert. : I am J. Smith
Serial Number
Attr. cert. : J. Smith is a manager
Source Of Authority

Figure 2- ACs and PKCs Differences

In Figure 2, the difference between these two types of certificates has been detailed. The issuer of a PKC[3], called a CA[4] in the PKI structure, issues and signs these certificates. On the other side, the AC issuer in Privilege Management Infrastructure is called Source of Authority. In the above figure, a hypothetical person has a PKC that authenticates him and has been issued by CA; and on the other hand, he has an AC that confirms his role as the "manager." In this paper, the user refers to the XCloud console and presents the AC to register his specifications in the system[7]. We will discuss how this console is designed later in the paper.Managing these Attribute Certificates and performing tasks like issuance, delegation, revocation, etc. requires a management infrastructure and related entities to be defined. The PMI theory has been presented for this purpose. The most important application of this infrastructure is "confirmation of the features defined for the certificate holder." This structure is responsible for management of Attribute Certificates by defining entities and defining certain obligations for each[8].

[3] Public Key Certificate
[4] Certificate Authority

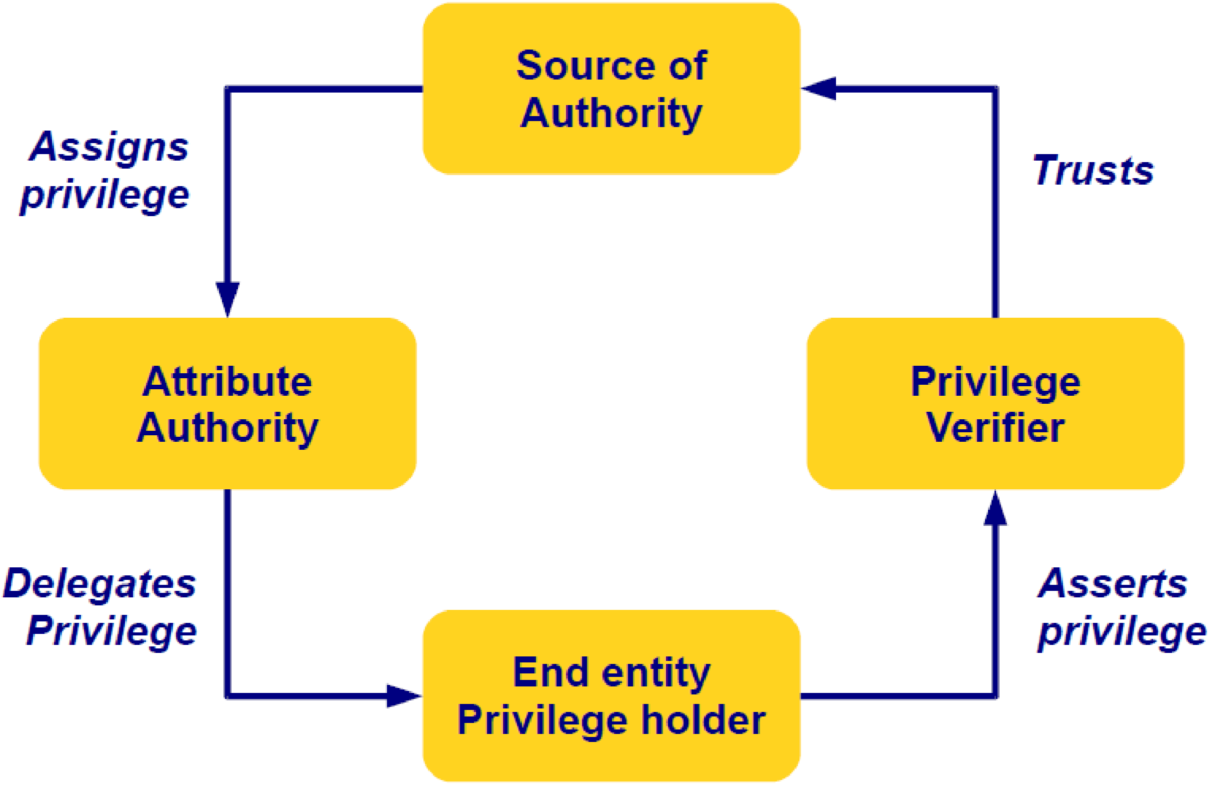


Figure 3- A PMI key elements

As displayed in Figure 3, four entities are defined in this structure:

1- Source of Authority (SoA): This entity acts as the root, and all trust it by default. This trust is applied to certificates issued through private key signature.

2- **Attribute Authority (AA)**: This entity, which can exist in any number, is allowed by SoA to issue certificates as well as to delegate them to others.

3- Privilege Verifier: This entity confirms the originality of a certificate by verifying the trust chain.

4- **End User**: This entity is recognized as the user or certificate holder.

In the XCloud framework, we define a SoA and as many AAs as there are sites, and for the sake of simplicity of the design, we do not use the issue of delegation. However, in actual environments and through establishment of delegation rights for others, this feature can make the certificates issuance procedure much more efficient considering all aspects.

2-3. Related work

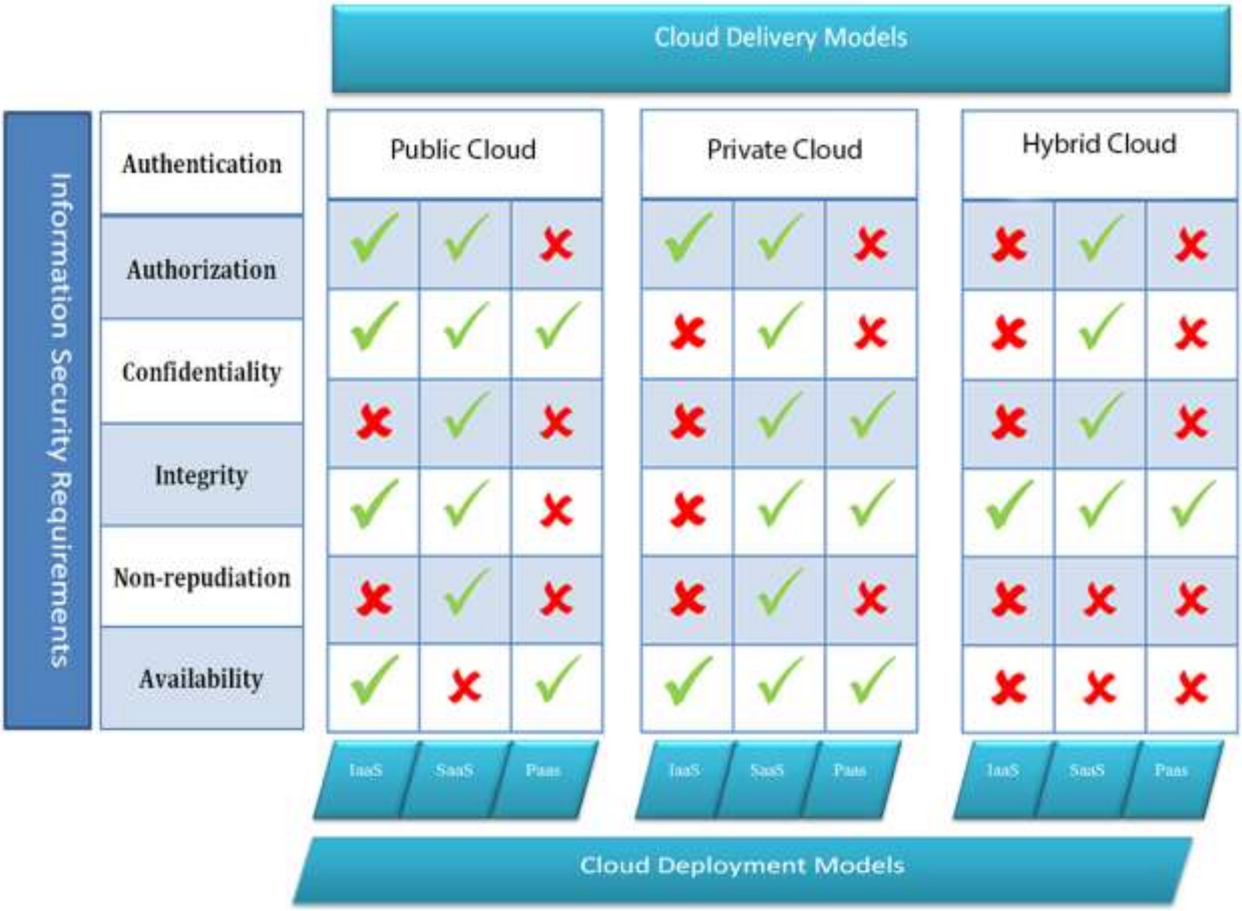
ADAMS is an instance of efforts made to increase security and performance in the area of cloud computing using PMI. Using OAuth[5] and introducing ADAMS, attempts to define a

[5] Open Authentication

system using ACs in order to increase the security of cloud computing service providers. ADAMS simplifies the available authentication methods of the complicated SSL or PKI structures[9][10].Another instance of these efforts comes in this reference[11]. This research has presented dCloud as a framework for IaaS service providers. This framework has been designed to decentralize authorization and access control in distributed environments. This system simplifies granting of resources to users using RBAC[6]. dCloud has not made use of PMI, and has found it sufficient to use RBAC, one of the concepts presented by PMI[12].

2-4. Rationale of the study

The studies introduced in Section 2-3 reveal more than ever the necessity of introducing a system offering the advantages of these systems simultaneously. In XCloud, we use PMI, which offers the authentication and authorization services simultaneously, and besides, we will introduce a management system as a part of XCloud to attempt to improve the structured, decentralized method of client management. To further clarify the issue, we first examine the security requirements in cloud computing environments.


Cloud Delivery Models
Information Security Requirements
Cloud Deployment Models

	Public Cloud			Private Cloud			Hybrid Cloud		
	IaaS	SaaS	Paas	IaaS	SaaS	Paas	IaaS	SaaS	Paas
Authentication	✓	✓	✗	✓	✓	✗	✗	✓	✗
Authorization	✓	✓	✓	✗	✓	✗	✗	✓	✗
Confidentiality	✗	✓	✗	✗	✓	✓	✗	✓	✗
Integrity	✓	✓	✗	✗	✓	✓	✓	✓	✓
Non-repudiation	✗	✓	✗	✗	✓	✗	✗	✗	✗
Availability	✓	✗	✓	✓	✓	✓	✗	✗	✗

Figure 4- Cloud Computing Security requirements

As examined in Figure 4, cloud computing environments have different security requirements depending on the serving model and the access model[13]. In this study, we have introduced XCloud as a framework for IaaS serving environments and the public access model, and have embedded the required security requirements in this framework. To make a comparison to the

[6] Role Based Access Control

efforts made so far to design frameworks for cloud computing, we should specify criteria for comparison. These criteria include:

- comprehensiveness: that is, the presented framework should present security, management, and executive solutions simultaneously;
- use of X.509: this standard is in use as the standard accepted in today's world of Internet; use of this standard means easier development capabilities and easier communication with infrastructures available in the world of Internet;
- decentralization: this capability means the ability to apply the framework in question to geographically distributed environments, without disturbing the serving quality;
- information security: due to criticality, we have used 2 different types of definition in this criterion:
    - intra-structure security: that is, use of different encoding protocols and different security modules in order to provide security within a work node; for example, modules that operate within a host server to monitor operations of other virtual machines, and prevent malware from expansion from one virtual machine to another machine; or prevent virtual machines from access to other virtual machines' information;
    - Extra-structure security: capabilities like authentication, access control, and authorization defined as extra-structure security requirements.

	comprehensive design	Based on X.509	Decentralization	Security	
				Extra-structure	Intra-structure
XCloud	✓	✓	✓	✓	✗
ADAMS	✗	✗	✗	✓	✓
dCloud	✗	✗	✓	✓	✗

Figure 5- Comparison between frameworks

In Figure 5, studies introduced so far in the area have been compared from the point of view of the criteria introduced above. In regard to the intra-structure security in XCloud, it should be noted that the VMWare virtualization software meets this requirement to some extent, and it is among the future research purposes of the XCloud framework to implement this class of capabilities.

2-5. Challenges and security issues of cloud computing

Security is a major challenge in cloud computing. Physically and logically, data can be stored anywhere in cloud computing. The user does not need to know where exactly his personal information and settings are accessed[1]. However, consideration of security and secrecy issues in cloud computing requires that we assure the client that these operations are under the regulations and agreements formed and do not violate the user's privacy.

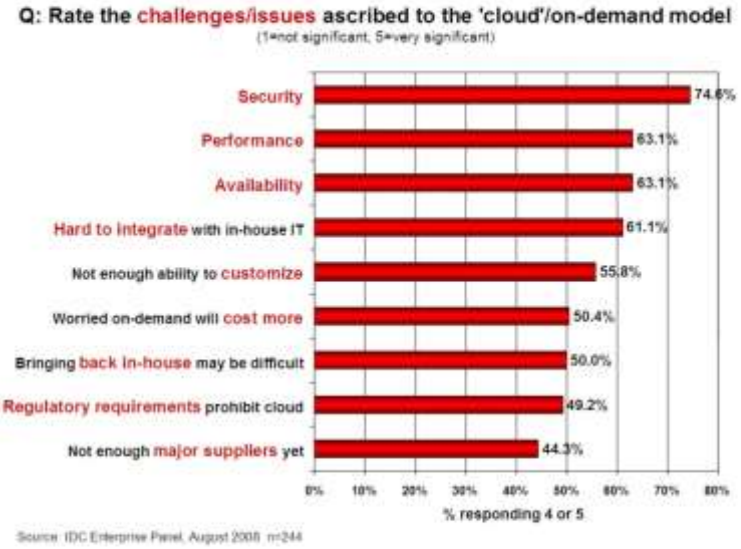


Figure 6- IDC cloud challenges survey[14]

Figure 6 displays the present challenges of cloud computing, examined by IDC in August, 2008. Besides the positive features mentioned before, security issues and availability are the most important challenges facing this field. A wide range of standards and organizations are operating in the field of cloud computing security, the major ones being stated below[14].

- Cloud Security Association (CSA)
- The Standard Series ISO 27001-27006
- European Network and Security Agency (ENISA)
- Information Technology Infrastructure Library (ITIL)
- National Institute of Standards and Technology (NIST)

Nevertheless, many organizations and institutes of standard have designed their jobs based on their own needs. But investigation of these standards helps us pay attention to these security considerations in the proposed architecture.

3. The proposed architecture and an introduction to XCloud

3-1. System structure

In the same way as we introduced XCloud in the Introduction section of the paper, we now examine the architecture proposed for this system. The XCloud framework is composed of 2 structural sections: clients' portal and processing engine. In the operation procedure, the framework receives a certificate from the user via the clients portal part, and after authentication of the user and specification of access levels, it sends this information toward the processing engine. In this part, the virtual machine is created using the APIs used, and the information about the virtual machine made is sent to the user.

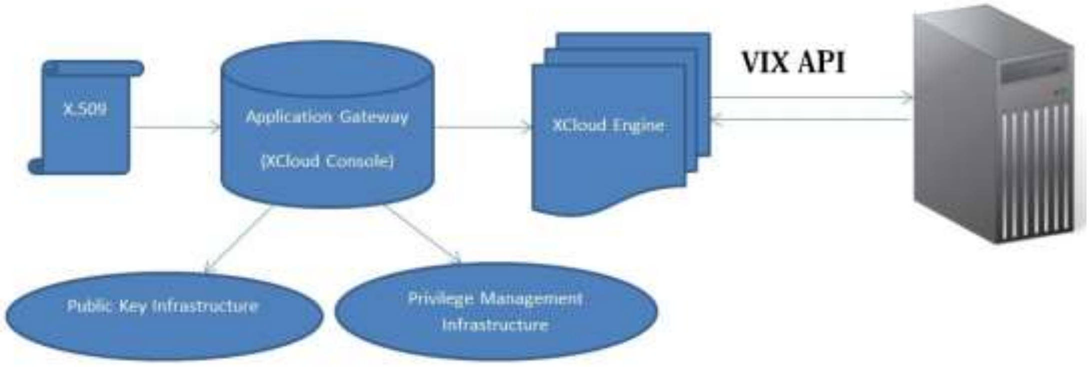

X.509
Application Gateway
(XCloud Console)
XCloud Engine
VIX API
Public Key Infrastructure
Privilege Management Infrastructure

Figure 7- Basic Diagram of XCloud Structure

Utilizing X.509 ACs and using virtualization platforms, XCloud has presented a modern solution in client management. In solutions based on traditional methods, usernames and passwords are usually used for authentication of users. In this method, use is made of an access list to control clients' accounts; but in the method presented in this paper, clients carry their accounts inside their certificates. Here, the client presents his certificate by referring to the application console, operating as the application gateway. Then, the application code creates the client account in the application database based on the certificate. At the beginning of the paper, we introduced the PDP and PEP entities in an authorization system. In this framework, the application gateway plays the role of PDP and the main application code acts as PEP. The communication between these 2 entities is established through the XACML language structure. After authenticated, the client's certificate is compiled in the console. As explained in the standard[7], in this certificate, a series of attributes can also be defined besides the general fields. We have made use of the "text" data type to express certificate features, where fields are defined as detailed below:

1- E-mail: specifications of the virtual machine made (IP address, username, and password) are sent to the address recorded in this field.

2- The number of CPU cores: this field has been specified in the certificate as CPUCount, and specifies the number of processing cores requested.

3- RAM amount: this field has been specified in the certificate as RAM, and specifies the amount of RAM requested in megabytes.

4- Disk amount: this field has been specified in the certificate as HDD, and specifies the amount of hard disk memory requested in gigabytes.

5- Network traffic: this field has been specified in the certificate as NetworkTraffic, and it restricts the amounts of information received and sent by this virtual machine.

6- Authorized network services: in this field of the certificate, in order to preserve security at the individualization level and prevent unauthorized access, the types of the services offered in the client's virtual machine are recorded.

3-2. Security at the virtualization level

[7] RFC 5755: An Internet Attribute Certificate for Authorization

Virtualization is considered as the main key technology in cloud computing. When the user presents his needs, he is assigned a virtual machine, and information on access to the virtual machine is offered to him. Because several virtual machines use a single host machine, this causes potential vulnerabilities to exist in the system. To prevent these vulnerabilities, some methods are suggested and examined later in the paper.

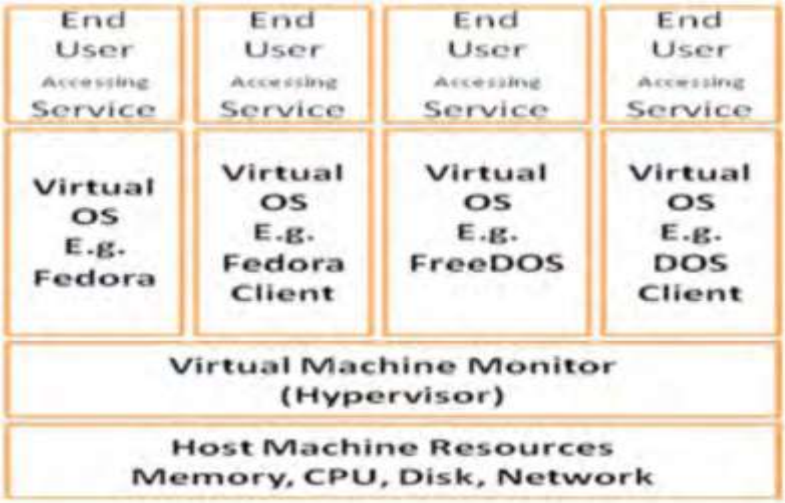

End User Accessing Service
End User Accessing Service
End User Accessing Service
End User Accessing Service
Virtual OS E.g. Fedora
Virtual OS E.g. Fedora Client
Virtual OS E.g. FreeDOS
Virtual OS E.g. DOS Client
Virtual Machine Monitor (Hypervisor)
Host Machine Resources Memory, CPU, Disk, Network

Figure 8- Virtualization Elements

Figure 8 displays how the virtualization technology is used for utilization in systems offering cloud computing services. This method of implementation of virtualization is based on security guarantees that the virtualizer software offers to us, and makes no further attempt[15]. Separation of virtual machines is one of the most important principles in implementation of safe virtualization. For this purpose, the following system is proposed to monitor the separation in XCloud. In this method, we use a virtualizer sub layer responsible for permanent monitoring of activities of each virtual machine. This automatic control device continuously performs operations such as scan of network ports, services, and protocols during the virtualization process, through which we can separately examine input and output requests for each virtual machine. This model is implemented using the VMM-Master module, introduced in SVM[16].

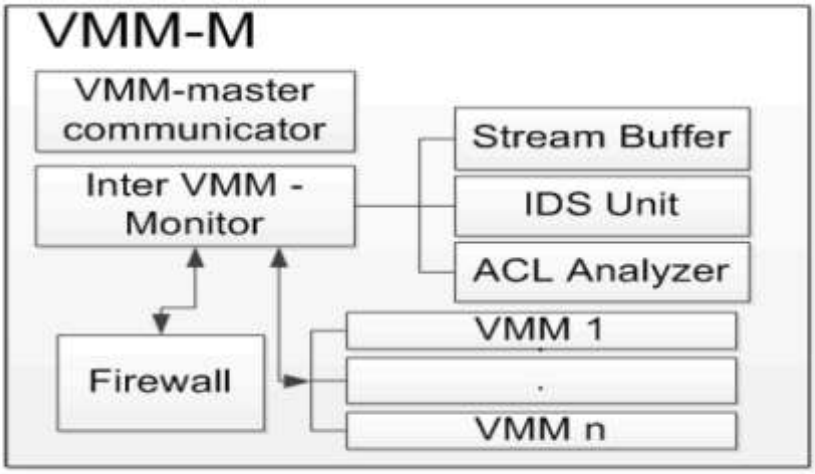


Figure 9- SVM Diagram[16]

As displayed in Figure 9, through well-timed monitoring of information from each virtual machine, this module discovers and frustrates attacks to virtual machines. As explained in Section 3-1 of this text, in certificates related to each client, the types of the requested services are presented. What is meant by services in this section is network services. For example, a virtual machine can be authorized only to offer web hosting an e-mail services. For this purpose, network ports required by this virtual machine are stored in the ACL section of VMM-M to make it easier to verify and identify any unauthorized operation.

3-3. The extended model

The above structure has been designed and implemented for the purpose of presentation of a single-point model of a cloud computing management system. Considering the requirements in actual environments, the need is felt for a design with desirable performance in distributed environments. In actual environments like the international network of Internet, service providers are required to be distributed in physically different points to offer such a service. To design such a model, we require a method for exchange of information on authentication and authorization among nodes distributed in the network. This architecture has been explained in Figure 7. For example, imagine a user who refers to Site A of XCloud and whose information on access level is confirmed. If the user has to move his virtual machine physically while using the service, say from one country to another, such a possibility has been considered in this model.

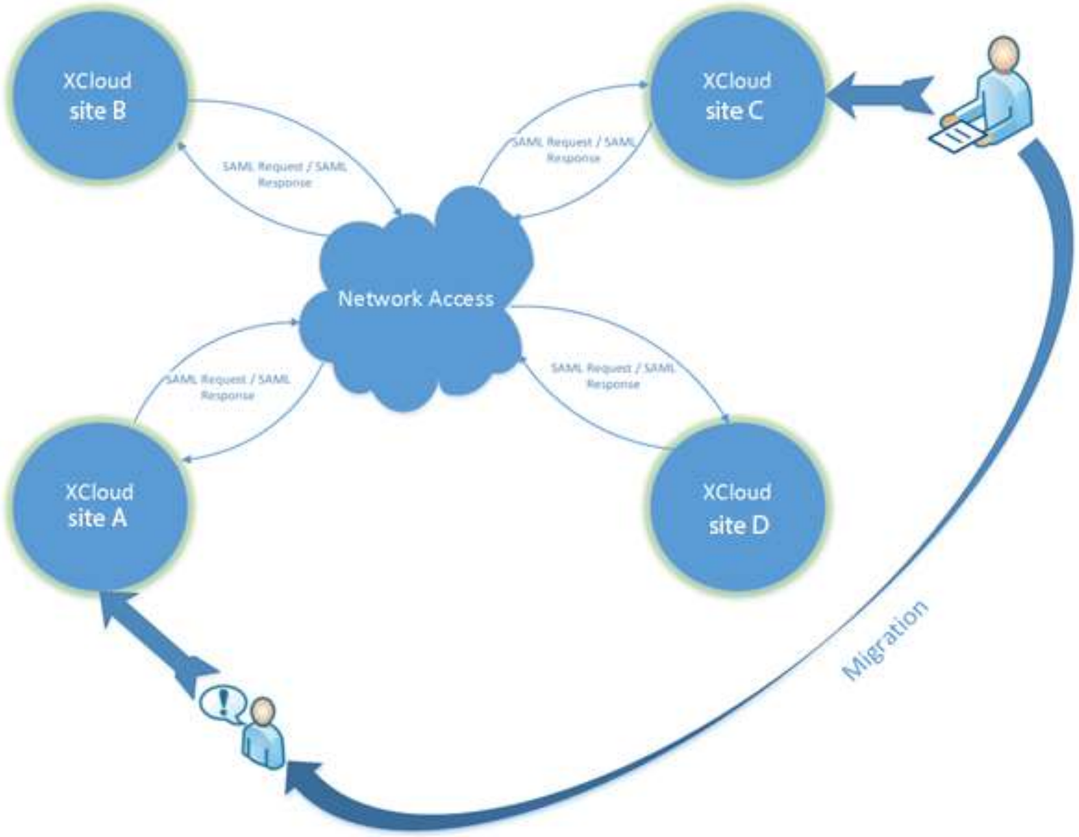

XCloud site B
XCloud site C
SAML Request / SAML Response
SAML Request / SAML Response
Network Access
SAML Request / SAML Response
SAML Request / SAML Response
XCloud site A
XCloud site D
Migration

Figure 10 - XCloud Extended Model Diagram

As shown in figure 10, for this purpose, we use the SAML[8] language structure to exchange this information. This language is a standard, open-source language based on XML, designed specifically for exchange of information on authentication and authorization. In this scenario, described in Figure 10, the client's information and control of his access levels is performed upon reference to Site 1. The SAML structure is a very efficient method in transfer of this confirmed information to another place. In this method, upon the user's reference to Site C, the client only presents the username and password in the XCloud system to the site. Then, Site D sends the SAML request toward Site A. Responding to this request, a SAML response is sent, containing the client's authorization features in that site. This model is known in the world of Internet as the Single Sign On model. This model is also interpreted as entering the system once and using it in several places. And one of the simplest methods of application of the model is the SAML structure, used in the XCloud framework. Two basic entities play roles in the SAML structure: service provider (SP) and identity provider (IdP). In the plan we present for distributed structures, each node should be able to play both roles mentioned, in such a way that each node is capable of processing SAML requests from and sending SAML responses to other nodes.

[8] Security Assertion Markup Language

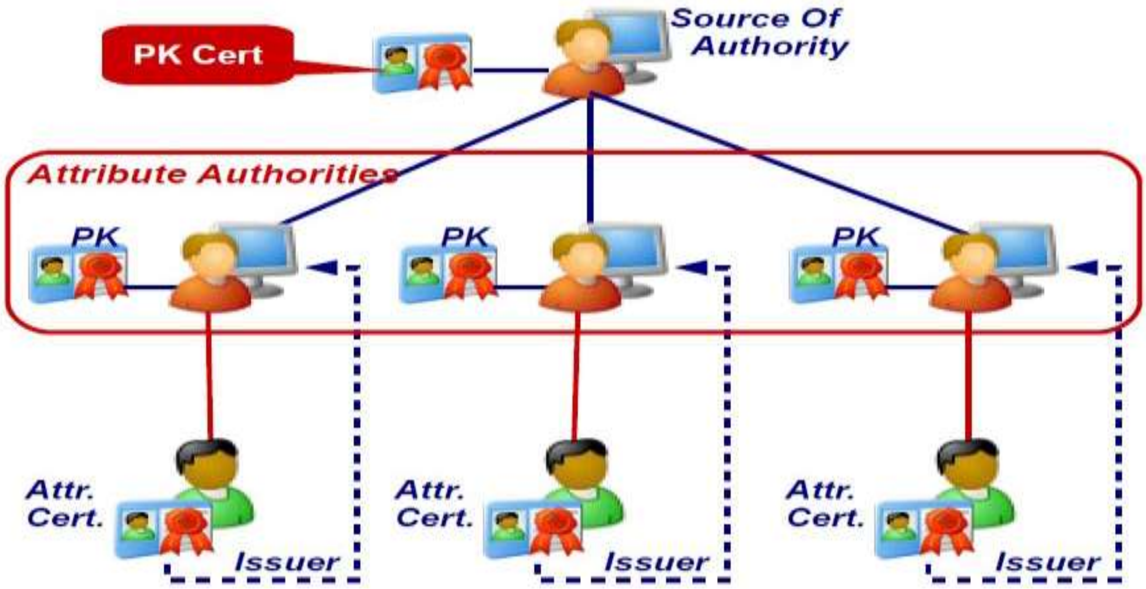
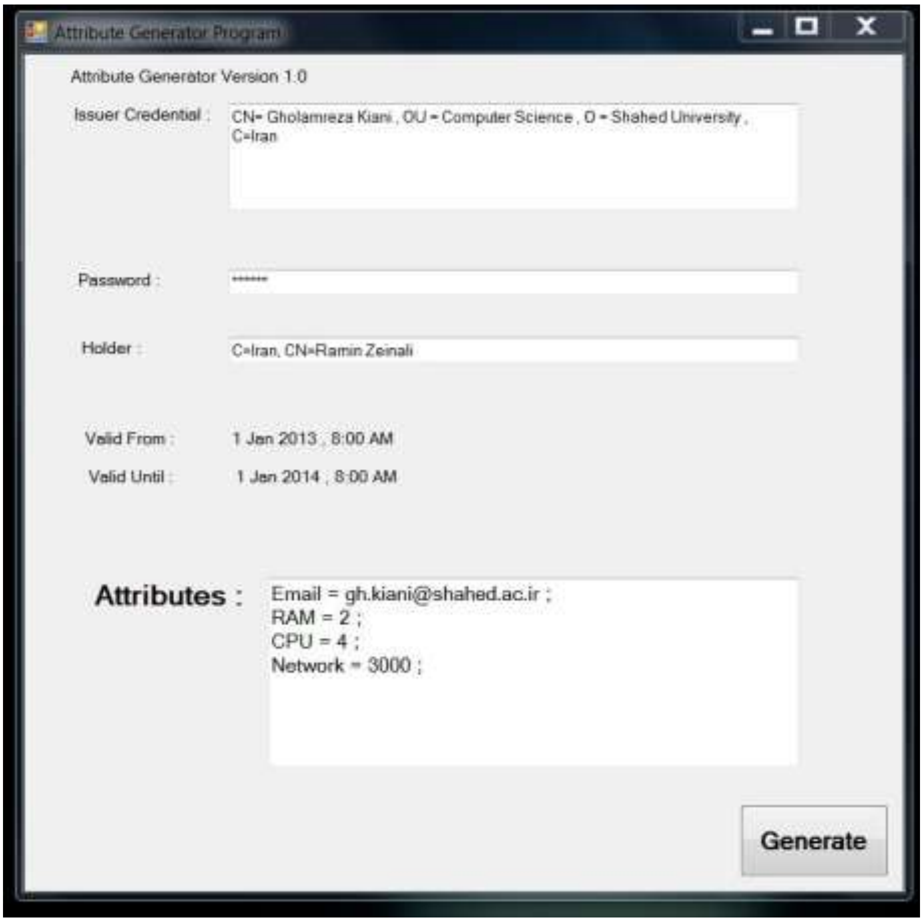


Figure 11- XCloud Management infrastructure roles

In order to run a PMI infrastructure in the XCloud framework, we define a SoA and as many AAs as there are sites. Figure 11 shows an example of this management infrastructure. As explained at the beginning of the paper, to create a trust verification structure, we also need a PKI infrastructure. That is, each of the SoA and AA entities require a PKI infrastructure to prove the originality of their public keys. For this purpose, use can be made of software designed to implement this structure. As displayed in the figure, all ACs must be signed by a single issuer. In the XCloud framework, these issuers are different sites, each of which are allowed to issue ACs with specifications of that service for their clients in return for the price receipt of the service required by the client. When these certificates are issued, the validity date of the certificate is also defined based on the service duration.

3-2. Implementation details

To implement this framework, we have used the C# programming language, and the application programming interfaces have been written in the C++ language. We write the operations related to each interface in the C++ language, and convert them to a run-time library file. For example, to turn on a virtual machine automatically, a C++ language code titled TurnOn.C has been written. To use the functions of the code in the main program, we provide its run-time library file titled TurnOn.DLL, and then use it in the program code.



Attribute Generator Version 1.0

Issuer Credential : CN= Gholamreza Kiani , OU = Computer Science , O = Shahed University , C=Iran

Password : ******

Holder : C=Iran, CN=Ramin Zeinali

Valid From : 1 Jan 2013 , 8:00 AM

Valid Until : 1 Jan 2014 , 8:00 AM

Attributes : Email = gh.kiani@shahed.ac.ir ;
RAM = 2 ;
CPU = 4 ;
Network = 3000 ;

Figure 12- A sample of implementation. Issuing a certificate

Figure 12 has been included as a sample of the forms implemented in this framework. In this form, specifications of the certificate holder as well as information concerning authorization through attributes can be defined, and this eventually leads to issuance of an AC. As observed in Figure 9, these defined values can be altered at any time. After receiving these values from the certificate, the console sends the request for creation of a virtual machine with the specifications given to the program's main code, and after receiving confirmation, it sends an e-mail containing the created virtual machine specifications to the client. The program code communicates with the virtualized server, which is on the WMWare Workstation platform, through application programming interfaces (APIs)[17]. These application programming interfaces meet the basic needs in communication with a virtual machine. These interfaces perform functions like making a virtual machine, turning it on or off, changing specifications, obtaining backup versions, and other basic operations of a virtual machine. Clients access virtual machines through communication consoles provided for this purpose and information sent to e-mails.

3-3. Advantages of the proposed architecture

One of the advantages of this framework is that its input interface is standard, being based on X.509 certificates, in such a way that this gateway can also be used as the communication gateway for other virtualization platforms. Another advantage of this system is comprehensive, integrated management of users, such that client management is performed in the certificate issuance stage, and each user uses his service by holding this certificate and referring to any of the sites providing services. In general, use of this modern method of authorization based on ACs has several advantages, of which we list the most important. To explain these advantages, we first deal with the problems with the present authorization devices and how they can be improved using these certificates. These advantages include:

1- Support for authorization in distributed environments: Any authorization method based on authentication and granting of access on the basis of access control lists has high chances of occurrence of errors in distributed environments. This high risk is due to this type of systems' natural need for keeping access control lists up-to-date and free from contradictions. It would be very difficult to preserve these 2 very important features on these lists, particularly when access alters within short periods—like a few seconds. Public key certificates would not help here either, because first, they have high validity, and second, they do not support keeping and presenting the certificate holder's features in an encoded form either. Now, it is felt clearly that we need a system providing us with these authentication services as a third party with guarantee for security of the data within a certificate. The framework presented in this paper has also been developed in order to preserve and present clients' features in the form of certificates signed by a credited person where the originality of the features has been confirmed.

2- Support for access delegation right: In a distributed environment with a large number of users, one of the most important features that can be offered is delegation of access rights to others. In a traditional authentication environment, this feature is not easily applicable. In the XCloud framework, different issuer sites can delegate all or part of their rights to others. This delegation can be performed through fields available in an AC, and it is quite safe from a security point of view, since the originality of the delegation is absolutely confirmed by the delegator entity's signature. This feature allows us to have high flexibility in issuing or extending certificates. For example, as its proxy, a web service can perform part of an issuer site's tasks quite safely.

3- Easy implementation: Since it can be integrated with available PKI devices, implementation of this framework will have a minimal need for changes in present infrastructures. All protocols and functions can make perfect use of the advantages of including features in a safe certificate.

4- Encoding of the data within the certificate: In current authentication systems, no standard method has been predicted for encoding access control lists and other items. Based on the latest version of the X.509 certificates, the XCloud framework supports encoding of certificates via public and private keys of issuers and holders of certificates. The standard for information representation in these certificates is the ASN.1[9] format. Due to the ever-increasing use of the XML language format in the world of Internet, many discussions are being followed over support for this type of format, and later versions of the X.509 standard will definitely support XML.

[9] **A**bstract **S**yntax **N**otation **O**ne

5- Use of SVM: In the proposed system, we have used the offered SVM architecture, which allows us to discover and frustrate attacks at the vulnerabilities level of virtualization. Use of this method also allows us to statistically monitor services offered in each virtual machine, so that we can thus identify and monitor unconventional services as well.

4. Conclusion

In this paper, XCloud has been designed for environments providing cloud computing. At the beginning of the paper, we made mention of advantages of using a privilege management system and ACs in implementation of an authentication system. Considering the practical aspect of implementation of this framework in cloud service provider environments, we can benefit from all advantages of such a system. Also, attention should be paid to the carful point that this system requires a comprehensive, fully-monitored management structure, like what has been implemented now in the public key infrastructure; therefore, precision in definition of key entities like SoA and AA has considerable effects on increase in the system effectiveness. Also, as mentioned, for the system to be effective in actual environments, most of which are distributed environments, use of safe methods of security information exchange has been predicted, the best at present being the SAML protocol. In the implementation made, an XCloud work node or site available for the research purposes has been implemented in practice. Finally, considering the management and security developments that can be considered for this structure, and considering the point that all the advantages suggested in the first section of the paper can entirely be obtained, this architecture can be used as the foundation for designing cloud computing service providers with the latest standards presented for digital certificates.

References:

[1]S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of cloud computing,” J. Netw. Comput. Appl., vol. 34, no. 1, pp. 1–11, Jan. 2011.

[2]C. Ruan and V. Varadharajan, “Dynamic delegation framework for role based access control in distributed data management systems,” Distrib. Parallel Databases, Jan. 2013.

[3]Q. Zhang, L. Cheng, and R. Boutaba, “Cloud computing: state-of-the-art and research challenges,” J. Internet Serv. Appl., vol. 1, no. 1, pp. 7–18, Apr. 2010.

[4]ietf RFC, “Internet X.509 Public Key Infrastructure Certificate,” 2015. [Online]. Available: https://tools.ietf.org/html/rfc5280.

[5]OASIS, “XACML v2.0 Core, eXtensible Access Control Markup Language Version 2.0,” OASIS, 2015. [Online]. Available: http://www.oasisopen.org/committees/xacml.

[6]D. W. Chadwick and A. Otenko, “The PERMIS X.509 role based privilege management infrastructure,” Futur. Gener. Comput. Syst., vol. 19, no. 2, pp. 277–289, Feb. 2003.

[7]J. Pascal, “Understanding X . 509 Attribute Certificates,” White Pap. , Syst. Archit., pp. 1–6, 2009.

[8]B. Blobel, P. Hoepner, R. Joop, S. Karnouskos, G. Kleinhuis, and G. Stassinopoulos, “Using a privilege management infrastructure for secure web-based e-health applications,” Comput. Commun., vol. 26, no. 16, pp. 1863–1872, Oct. 2003.

[9]J. K. Moon, H. R. Kim, and J. M. Kim, “Privilege management system in cloud computing using OAuth,” Int. J. Secur. its Appl., vol. 8, no. 3, pp. 221–234, 2014.

[10]OASIS, “OASIS Security Services (SAML).” [Online]. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security.

[11]D. Shin, H. Akkan, W. Claycomb, and K. Kim, “Toward role-based provisioning and access control for infrastructure as a service (IaaS),” J. Internet Serv. Appl., vol. 2, pp. 243–255, 2011.

[12]D. F. Ferraiolo and D. R. Kuhn, “Role-Based Access Controls,” 15th Natl. Comput. Secur. Conf. Balt. MD. Oct. 13-16, 1992, pp. 13–16, Mar. 2009.

[13]T. Dillon, C. W. C. Wu, and E. Chang, “Cloud Computing: Issues and Challenges,” Adv. Inf. Netw. Appl. (AINA), 2010 24th IEEE Int. Conf., pp. 27–33, 2010.

[14]“IDC , Research of information technology companies and markets. Free newsletters. Custom consulting services.” [Online]. Available: www.IDC.com.

[15]A. S. Ibrahim, J. Hamlyn-Harris, J. Grundy, and M. Almorsy, “CloudSec: A security monitoring appliance for Virtual Machines in the IaaS cloud model,” in 2011 5th International Conference on Network and System Security, 2011, pp. 113–120.

[16]S. Manavi and S. Mohammadalian, “Secure Model for Virtualization Layer in Cloud Infrastructure,” Int. J. …, vol. 1, no. 1, pp. 32–40, 2012.

[17]“VMware, Inc. is a U.S. software company that provides cloud and virtualization software and services,” 2015. [Online]. Available: www.vmware.com.