International Journal of Advanced Science and Research
www.allsciencejournal.com
ISSN: 2455-4227
Received: 25-08-2021; Accepted: 10-09-2021; Published: 25-09-2021
Volume 6, Issue 3, 2021, Page No. 08-14

Design and development of a uniqueness organization scheme using cloud for authentication and authorization

S Ravichandran[1], S Akila[2]
[1] M.Sc., Department of Computer Science, Shree Chandraprabhu Jain College, Minjur, Chennai, Tamil Nadu, India
[2] HOD & Assistant Professor, Department Computer Science, Shree Chandraprabhu Jain College, Minjur, Chennai, Tamil Nadu, India

Abstract
The emergence of cloud computing paradigm offers attractive and innovative computing services through resource pooling and virtualization techniques. Cloud providers deliver various types of computing services to customers according to a pay-per-use economic model. However, this technology shift introduces a new concern for enterprises and businesses regarding their privacy and security. Security as a Service is a new cloud service model for the security enhancement of a cloud environment. This is a way of centralizing security solutions under the control of professional security specialists. Identity and access control services are one of the areas of cloud security services, and sometimes, are presented under the term Identity as a Service. This master thesis research is focused on identity-security solutions for cloud environments. More specifically, architecture of a cloud security system is designed and proposed for providing two identity services for cloud-based systems: authentication and authorization. The main contribution of this research is to design these services using service-oriented architectural approach, which will enable cloud-based application service providers to manage their online businesses in an open, flexible, interoperable and secure environment. First, the architecture of the proposed services is described. Through this architecture all system entities that are necessary for managing and providing those identity services are defined.

Keywords: Object oriented concepts, integrated environments, dynamic analysis, race detection and software maintenance

Introduction

Identity and Access Control Service should provide identity management and access control to cloud resources for registered entities. Such entities can be people, software processes or other systems. In order to give a proper level of access to a resource, the identity of an entity should be verified first, which is the authentication process that precedes the authorization process. Besides authentication and authorization processes, audit logging mechanism should be used to keep track of all successful and failed operations regarding authentication and access attempts by the application [1]. Confidentiality is achieved by different encryption mechanisms, which is the procedure of encoding data by means of cryptographic algorithms.

Providing such a service will guarantee privacy of sensitive and private data and the intended entity can only decode it. Cryptographic algorithms, which are computationally hard to crack together with encryption and decryption procedures, digital signatures, hashing, certificates, key exchange and management form an encryption system which can be delivered as a service and assure confidentiality and non-repudiation in a cloud environment [1]. As such, the centralization of security services and implementation of those services through standardized security frameworks under the model of SaaS can be viewed as an innovative and beneficial utility for a cloud environment. This approach promotes the delivery of security services to customers in a professional and standardized manner [12]. Many motives can be pointed for such kind of solution for a cloud environment: 1-aggregation of security skills and security experts, 2-effective centralized solution, 3-standardization of security practices, 4- competitive advantage in the market over the competitors. The effective management of security in cloud-based applications is one of the core factors for the successful cloud computing platform [3]. Identity as a Service (IaaS) is one area of SaaS and it aims to provide security services within the scope of “identity eco-system” of a cloud environment [4]. Existing cloud-based identity service mechanisms require constant improvements and enhancements as identity associated security risks have become one of the most significant issues for a cloud environment. Privacy protection for identity information is critical factor for a successful identity system [5]. The contributions of this research will be within the area of identity services for cloud environments and will be focused on designing a cloud security system which addresses current identity-related security issues Identity and Access Control Service should provide identity management and access control to cloud resources for registered entities. Such entities can be people, software processes or other systems. In order to give a proper level of access to a resource, the identity of an entity should be verified first, which is the authentication process that precedes the authorization process.

Besides authentication and authorization processes, audit logging mechanism should be used to keep track of all successful and failed operations regarding authentication and access attempts by the application [1]. Confidentiality is achieved by different encryption mechanisms, which is the procedure of encoding data by means of cryptographic algorithms. Providing such a service will guarantee privacy

of sensitive and private data and the intended entity can only decode it. Cryptographic algorithms, which are computationally hard to crack together with encryption and decryption procedures, digital signatures, hashing, certificates, key exchange and management form an encryption system which can be delivered as a service and assure confidentiality and non-repudiation in a cloud environment [1]. As such, the centralization of security services and implementation of those services through standardized security frameworks under the model of SaaS can be viewed as an innovative and beneficial utility for a cloud environment. This approach promotes the delivery of security services to customers in a professional and standardized manner [2]. Many motives can be pointed for such kind of solution for a cloud environment: 1-aggregation of security skills and security experts, 2-effective centralized solution, 3-standardization of security practices, 4- competitive advantage in the market over the competitors. The effective management of security in cloud-based applications is one of the core factors for the successful cloud computing platform [3]. Identity as a Service (IaaS) is one area of SaaS and it aims to provide security services within the scope of "identity eco-system" of a cloud environment [4].

Related Work

Cloud computing offers on demand services to customers with the properties of distributed systems, such as unlimited virtual resources, dynamic scalability, as well as cost advantages for business organizations. Security issues that arise within this computing environment result in various obstacles from both business and technological perspectives. There is a continuous development of security solutions with lots of challenges for a cloud environment. Security as a Service is a rather new approach to provide security solutions for a cloud environment in a professional and centralized way. Because SaaS delivery model is very broad and not a concrete implementation and currently still in its improvement stage, few cloud providers have a system that contains centralized security infrastructure, which can provide all the needs of customers from the security perspective. Cloud- based IaaS is not a well-established practice and there is a big need of transparent and simplified cloud security infrastructure that will provide identity management services to cloud-based software services.

As a solution to this problem, it is use to manage authentication and authorization systems in cloud environments and offer an approach of cloud security system for providing authentication and authorization services to cloud based software services through IaaS model. At the same time, it As a solution to this problem, it is use to manage authentication and authorization systems in cloud environments and offer an approach of cloud security system for providing authentication and authorization services to cloud based software services through IaaS model. At the same time, it will focus to deliver those services in an interoperable and secure manner.

The main purpose is to achieve a solution that provides secure and inter operable authentication and authorization systems in a cloud environment. The goals of this master thesis are the following

Design security system architecture for a cloud environment, which aims to deliver two identity services, such as authentication and authorization in a secure and interoperable manner, using Web Service technology. This solution will assist cloud computing platforms to provide software services to customers in a confidential, authenticated and authorized environment.

Develop and deploy a prototype of designed authorization service that will contain the main important features and findings of this investigation. The designed security system ensures a secure and reliable environment for cloud-based application services which is very easy to deploy and exploit on cloud-based platforms.

Provide an approach to build cloud security system for ensuring identity management and access control solutions for cloud-based application service providers through open and platform – independent architecture

Invention

Invention of Service Interfaces

The proposed shared security system is designed using WS technology. All system entities in the shared security system act as service providers and deliver security related solutions to cloud-based entities. They have well defined interfaces which enable service requesters to consume those services without any complexity. Each service has an input parameter (some services may not require an input parameter) and corresponding output parameter. These parameters conform to request and response messages for each service. The request- response messages are wrapped into the XML format, thus making platform independent system entities to interact with each other in an interoperable manner. Each service provider has a description of provided services, which is remotely available to service requesters. The security service providers register their services and publish their WSDL URLs at the IDMS. The application service provider looks up for the desired service at the IDMS service provider and obtains the URL of the WSDL file for that particular identity service. Then the application service provider obtains the WSDL document and based on that description the service can be easily consumed.

SSO Service

SAML server provides a SSO service to application service providers. The end-user authentication process is completely controlled and managed by the central security system of a cloud environment. For this system all SAML messages are transmitted using the HTTP- Redirect or HTTP-POST binding. In order to get a SAML ticket, the PEP server needs to connect to SSO service provider endpoint for incoming requests and call the request SAML ticket service. Through this call it sends a SAML authentication request message to the SAML server. The message is directed to the SAML server through the central authentication server which acts as a proxy server. The latter intercepts the message. As the request message is for authentication purposes, it starts to authenticate the end-user. The authentication result and SAML authentication request message are passed to the SAML server. In turn, SAML server issues a SAML authentication response message based on the authentication result and request messages. The SAML authentication request message must contain assertion ID for a particular message, ID and service URL of the service requester in this case the ID and service URL of the application service provider. The ID must match the registered ID in the IDSM database and service URL must

match the one described in the service metadata.
The message also contains assurance level of identity parameters for the authentication process: identity verification will be at that level. There may be other elements included in the request message. At the end, the message should be digitally signed by the service requester. The authentication result contains the subject ID according to the requested format and the status code and value of the identity verification process. The SAML authentication response message must contain assertion ID of the request message, ID and service URL of the SSO service provider, authentication result status, and assurance level of performed identity verification. There may be additional elements included in the response message. Before sending back the response message, it should be digitally signed by the SSO service provider.

SSO Service Protocol
Any user or client application, before accessing any resource provided by the application service, is first required to be authenticated. The SSO that can be IdP-initiated or RP-initiated in our system and the authentication process contains multiple interactions between different system entities. Figure 1 shows communication protocol between participating entities in the SSO service. This is a RP-initiated SSO. The end-user first connects to the application service provider through request resource message in order to request access to a protected resource or service. The request message is intercepted by the PEP server. If the end-user does not have a valid local session for that particular application service, PEP returns an authentication request message, such as SAML authentication request and directs the end-user to the SSO service provider.
The user connects to the strong authentication server via HTTP Redirect message protocol. Then the authentication process is executed according to the Strong Authentication Protocol provided by FIPS 196 specification. In the protocol diagram the authentication server authenticates only the end-user and it is based on the user's X.509 certificate. In some cases a user may also authenticate the authentication server. Then the user's identity registration is verified with the IDMS service provider. Besides, the authentication server communicates to the LCA server in order to check the validity of user certificate against certificate revocation list published by the LCA service provider. Getting the certificate verification result, the authentication server requests the SAML server to issue a SAML ticket. The SAML authentication response ticket is returned to the user through the authentication server according to the HTTP Post message protocol. Then the user is redirected to the application service provider. The response message is intercepted by the PEP, which verifies first the ticket validity and then may grant the user access to authorized resources or services based on that SAML ticket. More specifically, if the user has been successfully authenticated, then PEP creates a local valid session.

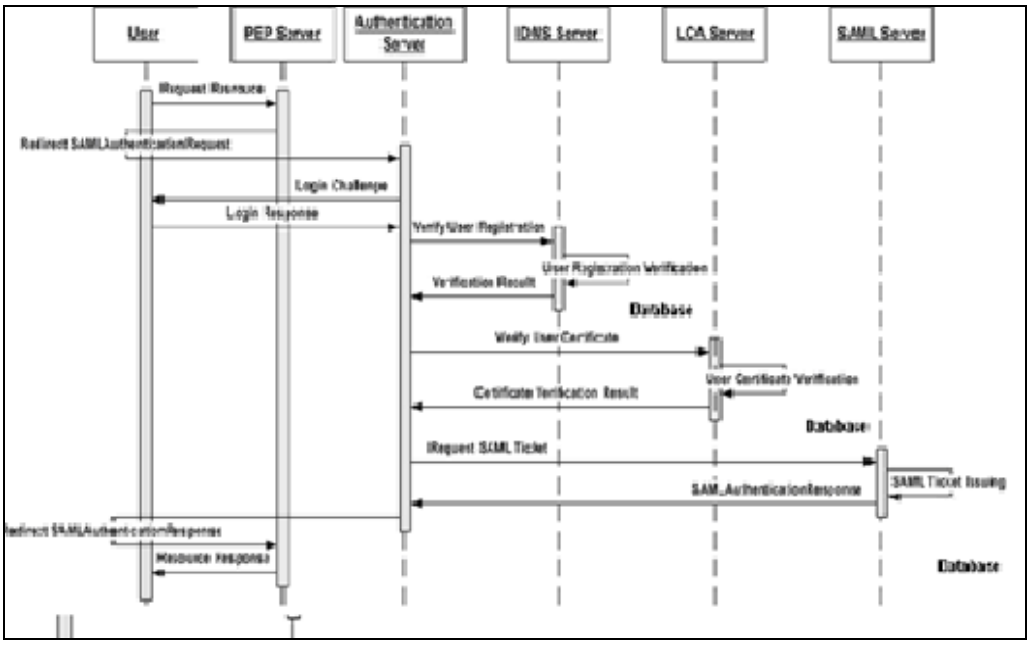

Fig 1: Block Diagram for SSO Service Protocol

In case of any SAML ticket issuing failure, the service returns a response message containing the failure information together with the failure status code.

Authorization Service
After successful authentication, the user may request protected resources or services. As described in previous chapter, central security system is responsible for authorization decisions as well. The PDP server delivers a single service which provides authorization decision based on XACML policies. The service requester (PEP) needs to connect to the PDP service endpoint, obtain a reference to the service object, and call the Request_XACML_Authorization_Decision service. Through this call PEP server communicates with PDP service provider using authorization decision request and authorization decision response messages, which are in XACML format. As mentioned in Chapter 3, we have adopted SAML-conformant PEP for our designed security system. Therefore, authorization decision request and authorization decision response message are embedded in SAML authorization request and messages. Because PDP service provider makes authorization decisions based on policy files in XACML format, there is a need to map each

SAML authorization request message into XACML request context and XACML response context into SAML authorization response message. In order to map and transfer XACML- formed request-response messages in SAML- based messages, SAML profile of XACML should be used. The complete specification of this profile can be found in the following referenced document [45]. It is the Pep's responsibility to protect resources from incoming requests and initiate an authorization evaluation process. The SAML authorization request message must contain assertion ID for a particular message, ID and service URL of the application service provider. The ID must match the registered ID in the IDSM database and service URL must match the one described in the service metadata. The message should be digitally signed by the PEP. The SAML authorization response message must contain assertion ID of the request message, ID and service URL of the PDP service provider. Before sending response message to the PEP, it should be digitally signed by the PDP service provider:

Authorization Service Protocol

PEP must control access to different application services, when user or any other system entity requests access to protected resources or services from the application service provider. Figure 2 shows the communication protocol for the authorization service. The user requests access to a resource at the application service site. PEP server intercepts the request message and constructs a SAML authorization request message including XACML authorization decision query statement which contains the requested resource URL, action and the role of the user. As it is role-based authorization system, each application service administrator must assign roles to users. As already mentioned in previous chapter, PAP provides user role assigning service and IDMS provides user attribute retrieving service. In order to obtain user role PEP must query IDMS. SAML authorization request message is sent to the PDP service provider. Upon receiving the message, the PDP service provider makes the authorization decision for that particular request and returns the XACML authorization decision result back to the PEP in a SAML authorization Response message.

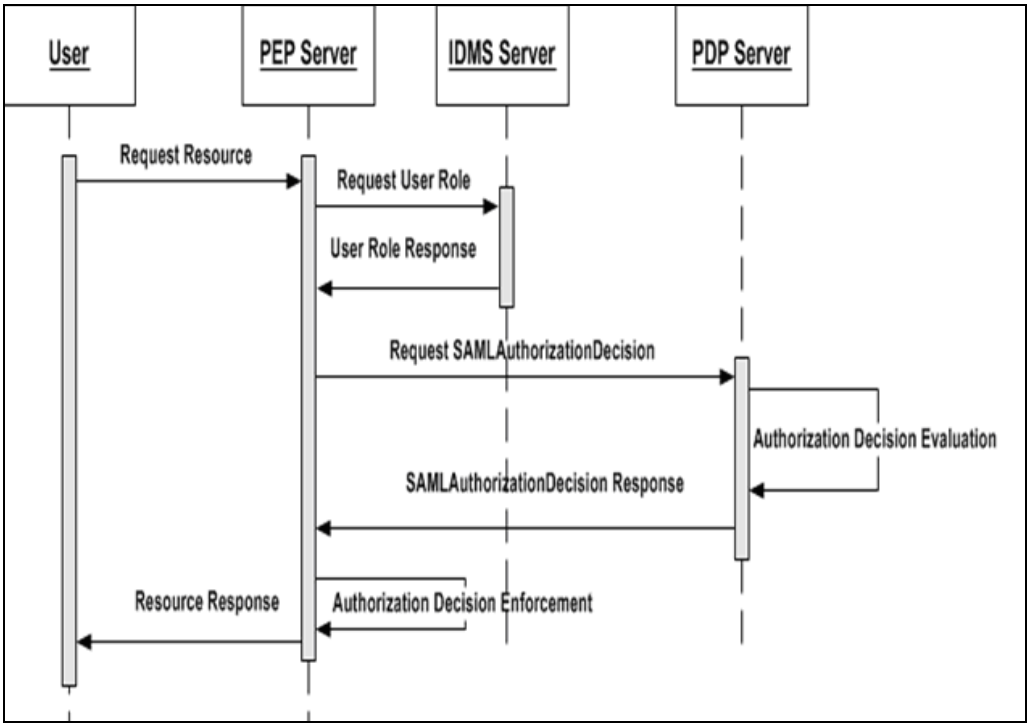

Fig 2: Block Diagram for Authorization Service Protocol

The message contains the decision status code and one of the four XACML decision values: Permit, Deny, Indeterminate or Not Applicable. If authorization decision evaluation is successful, meaning that PDP has positioned only registered policy for a particular XACML request, then the result contains target rule effect, such as deny or permit, defined by the security administrator. If there is no applicable policy for a particular XACML request (role, resource URL and action), then the result contains a Not Applicable decision value. In case of any XACML authorization decision issuing failure, the service returns the result as indeterminate together with the failure status code. Indeterminate decision value is also returned when PDP has located more than one policy for a particular XACML request. If the same policy contains two identical rules (even if rule effects are different), then the authorization decision result is evaluated against the first encountered target rule. In addition, the response message may contain an obligation or advice element. The PEP enforces the authorization decision: it either permits or denies the access. In case of permit, the application service returns the requested resource

Implementation

The implementation of the authorization system consists of two parts: Policy Administration Point (PAP) Service and Policy Decision Point (PDP) Service. The PAP service manages a role-based access control mechanism for security administrators and based on that service, the PDP service manages an authorization service for cloud-based system entities. The authorization service model is designed completely through the SOA approach; specifically using SOAP based Web Service technology. The execution is built on Java platform using already existing and tested

libraries and software frameworks. Applying Web Service technology to our authorization service implementation, it makes the system interoperable at a high level together with the ease of deploy ability, usability and system integration. Besides, Web Service technology provides all the necessary security mechanisms in order to manage secure environment between service requesters and providers.

Policy Administration

The implemented PAP service is used by system security administrators for managing and administrating role-based access control policies for their application service environment. It provides user friendly web interface for policy administration which is implemented using Java Server Faces (JSF) Model-View-Controller web framework and deployed in Apache Tomcat and My SQL servers.

The Apache Tomcat service provides runtime environment, more specifically, a container for the PAP service provider to deliver policy administration services to security administrators through http/https protocols. The My SQL database is used to store roles, rules, policies and other objects registered by administrators. Registered policies are also stored in the form of XML policy files.

The Hibernate Java persistent framework is used for managing and querying the database server. Security Administrators can register, view, update or delete objects such as roles, rules and policies. These actions conform to the four basic functions of persistent storage, such as create, read, update, and delete (CRUD). Besides, they can assign roles to already registered users and register application information. This PAP prototype implementation allows only registering one policy per each resource.

PAP Service access

Before accessing any service provided by the PAP system, the administrator must authenticate himself to the PAP service provider. Figure 3 shows the login page administrators are authenticated using username and password credentials which are defined and distributed to them beforehand in a secure manner.

Browser is used as a client application to connect to the PAP service via the Internet, so the communication channel is secured using SSL over HTTP, through which the service authenticates itself to the browser. This means that all the messages that are transmitted between the browser and PAP service provider are secured: SSL ensures the integrity and confidentiality of the messages transmitted between authenticated end-points, such as the browser and PAP service. After successful authentication, the security administrator can use all the services that are provided by the PAP system.

Creating and Managing policies

As the authorization service is based on the role-based access control mechanism, administrator can register a role using a web interface. Besides, already registered roles can be updated deleted. Figure 4 shows role registration panel: name, first, selects the “Roles” object on the left side of the web page and then the “Register” function. The role has three properties: name, description and domain. The role domain uniquely identifies the role within the scope of its usage.

If the administrator selects “List” function, all registered roles are listed. Furthermore, there are two corresponding functions for updating and deleting purposes. It shows, when “List” function is selected, it displays all the registered roles which can be then updated or deleted. There are nine roles which are defined and hard coded in the system beforehand for our cloud environment and during system initialization these roles become registered. Policy files contain role information for our role-based authorization system that is why it is necessary to register roles before defining policies.

If “List” function is selected, the list of registered rules are displayed which may then be updated or deleted. After defining the roles and rules, the administrator may register policies. Upon policy registration, an XACML policy file is created, digitally signed by the PAP entity using its private key and stored in a secure repository. For this implementation, policy files are not encrypted, although in some cases it may be required In order to register policies for the authorization system, “Policies” object should be selected. Again, “Register” function button displays the registration form.

The administrator gives the name of the policy, which should be unique; otherwise, it will not be stored in the repository. The value of “Rule Combining Algorithm” attribute should be one of the three given options. Each policy may have a “Condition” element that narrows the scope of policy evaluation. As we have a role-based authorization system, the value of the “Condition” element is one of the already registered roles which can be selected from the drop-down list. The “Resource URL” element defines the target of the policy file for which it will be evaluated.

Assigning Rules

The PAP service can assign already defined roles to registered users, because it shares a repository with IDMS for storing, updating or deleting user attributes. This prototype system does not allow assigning more than one role to user.

Application Registration

Generally, IDMS is responsible for providing application service registration services. However, the available implemented IDMS does not provide such service at the moment of this prototype implementation. When “Applications” object is selected on the left, two functions are enabled: “Register” and “List”. Figure 9 shows application registration panel. Calling “List” function, all registered applications are displayed, which can be read, updated or deleted.

Authorization service PDP

The PDP service is implemented using SOAP-based Web Service technology on Java platform. XACML open source Java libraries provided by J-Boss community are used in this PDP service implementation. The PDP service is running inside a Tomcat container as a runtime environment. The Web Service approach makes authorization service requesters, such as PEPs, very easy to consume it. PDP service description, as a WSDL file, is published in the same container. WSDL file of this service can be found in appendix A. Using this WSDL file, the PEP entity can easily connect to the PDP service provider and request the authorization service. This PDP service provider makes accessible only one service.

Request_XACML_Authorization_Decision

The service requires an XACML request message that contains three attributes: resource, action, and role and produces an XACML response message, that contains the authorization decision together with the corresponding status code. For this prototype implementation XACML request-response messages are not mapped into SAML based message protocols. Both XACML request and response messages are embedded in the body of SOAP messages.

SOAP messages are serialized in the XML format during the transmission. In order to make an authorization decision evaluation, PDP service loads all the registered policy files from the repository. Before evaluating any decision, each and every digital signature of the policy files is verified by the PDP service provider using the public key of the PAP entity. If the digital signature verification fails, the policy file will not further be considered for the evaluation. Thus, the authorization decision evaluation is based on policy files that have not been modified after originally created or updated by security administrators.

The communication channel between the PDP service and PEP entities is also secured though enabling the web service security mechanism called Mutual Certificates Security. This guaranties mutual certificate-based authentication between the PDP service and PEP entity, as well as message integrity and confidentiality exchanged between them. The service description includes also all the security related information that is required from service requesters in order to establish a secure communication.

Results and Discussion

The overall evaluation of the proposed security system from two perspectives: integration and security. Integration demonstrates how the proposed security services can be integrated within a cloud environment. Security demonstrates how securely the services are delivered to service requesters.

Advantages of web service security and integration

Both SSO and authorization services are designed using Web Service technology. Cloud computing platform is completely service- oriented and is accessed through high level Web API. That is why the integration of these security services within a cloud environment does not cause technology incompatibility issues. Moreover, it can effectively be deployed and exploited through utilizing all the benefits of service-oriented architecture. Here are the main Web Service advantages that the proposed cloud security system obtains from the service-oriented architectural design.

1. Loosely coupling
2. Standardized protocols
3. Interoperability
4. Usability
5. Deplorability

Result of system security

Security evaluation is based on the attack-oriented threat model. Threat model gives a formal approach to order potential security issues that makes the system security evaluation easy to understand. The proposed security system is analysed for possible security threats, taking into account security considerations for both authentication and authorization systems, highlighted in the previous chapter. There are five defined possible attacks for both services-replay attacks, message information disclosure (confidentiality), message modification, impersonation, and repudiation.

Table 1: System Security Evaluations

Security Services In	Security Threats				
	Replay Attacks	*Message Information Disclosure*	*Message Modification (Tampering)*	*Impersonation*	*Repudiation*
Authentication	Yes	Yes	Yes	Yes	Yes
Authorization	Yes	Yes	Yes	Yes	Yes

The above Table 1 shows whether both services are protected against those security threats. Replay attacks can be prevented using randomly generated session IDs (assertion IDs) in messages for both services. Message confidentiality is also protected for both services: in case of authentication service, messages are transmitted over a secure channel, such as SSL/TLS and in case of authorization service, messages can be encrypted using XML encryption standard. Message modification and repudiation are prevented using XML digital signature standard for both services. Impersonation attack is also prevented for both services using XML digital signature standard, as it can also provide information that the message is originated from intended entity.

In this research a cloud security system has been designed for managing authentication and authorization services applying quite new cloud service paradigm, such as Security as a Service. The proposed system supports deliver y of only two identity services. Therefore, more identity service features can be added, such as single log out, session refreshment, etc. The prototype implementation has some limitations: user can be assigned only one role at a time, there is no policy set concept applied for this system, and there is no separately implemented Policy Information Point service. Therefore, more features can be added to the prototype authorization system. Besides, a prototype of authentication system can be implemented according to the designed system. In this research a cloud security system has been designed for managing authentication and authorization services applying quite new cloud service paradigm, such as Security as a Service. The proposed system supports deliver y of only two identity services. Therefore, more identity service features can be added, such as single log out, session refreshment, etc. The prototype implementation has some limitations: user can be assigned only one role at a time, there is no policy set concept applied for this system, and there is no separately implemented Policy Information Point service. Therefore, more features can be added to the prototype authorization system. Besides, a prototype of authentication system can be implemented

according to the designed system.

Conclusions

Through this research a solution is provided for building cloud-based identity services, such as authentication and authorization based on the cloud SaaS model. This solution aims to provide an open and platform- independent architecture of a cloud security system, which is completely service-oriented, thus enabling the system to be scalable, interoperable, loosely coupled and location transparent.

Acknowledgments

The authors are thankful to Y. Demchenko, G.Ammons, and A. M. Ahmat for providing the necessary facilities for the preparation of the paper. Also thanks to IJASR Journal staffs to publish this paper.

References

1. Demchenko Y, Ramgovind S, Ziegler W. “Security Infrastructure for On-demand Provisioned Cloud Infrastructure Services,” in 2011 IEEE Third International Conference on Cloud Computing Technology and Science (Cloud Com), 2020, 55-263.
2. Stojčev MK, Kosanović MR, Golubović LR. Power Management and Cloud Energy Harvesting Techniques for Wireless Sensor Nodes. 2009 9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services, Nis, 7-9 October 2019, 65-72.
3. Cloud Security Alliance, “Security as a Service: Defined Categories of Service.” 2019.
4. S Ramgovind MM. Eloff E Smith. “The management of security in Cloud computing,” in Information Security for South Africa (ISSA), 2019, 1–7.
5. M Ates S, Ravet AM. Ahmat J Fayolle. “An Identity-Centric Internet: Identity in the Cloud, Identity as a Service and Other Delights,” in 2019 Sixth International Conference on Availability, Reliability and Security (ARES), 2019, 555–560.
6. Bharti A, Devi C, Bhatia V. Enhanced Energy Efficient LEACH (EEE-LEACH) Algorithm Using MIMO for Wireless Sensor Network. 2019 IEEE International Conference on Computational Intelligence and Cloud Computing Research (ICCIC), Madurai, 2015, 2019, 1-4.
7. Smitha Sundareswaran, Anna C, Squicciarini, Dan Lin, “Ensuring Distributed Accountability for Data Sharing in the Cloud”. IEEE Transactions on Dependable and Secure Computing,2019:9(4).
8. A Pretschner, Schuo F ¨ TZ, Schaefer C, T Walter. “Policy Evolution in Distributed Usage Control,” Electronic Notes Theoretical Computer Science,2020:244:109-123.