An Security Architecture for Campus Virtual Organization Using Role Based Access Control

M.Nithya[1] and Vijaykumar Varadarajan[2]

[1.] Associate Prof & Head, Dept of CSE,V.M.K.V.Engg College, SALEM, INDIA.
[2.] Professor, Department of CSE, S.A. Engineering College, Chennai, India
nithya.ph.d@gmail.com

Abstract: Many research efforts are being undertaken to improve shared resources security between Virtual Organizations (VO) in a grid environment, but user security is threatened as organizations in VO keep changing dynamically. VO access should be established only through trust relationships between VO and local users. Generally, there is no trust relationship between classical organization and VO or external members. Access to VO resources by organizations users is only when they possess a Certifying Authority (CA) provided certificate. Additional computing overheads increasing computational overheads are a disadvantage of using public-key for authentication. This paper proposes a novel authentication mechanism through the use of a public key concept being generated from the local Community Authorization Service (CAS) and not CA for increased speed. The proposed architecture accesses resources with dynamically generated keys and a token instead of signing in with credentials for every access by users. Simulation results prove that VO performance improves with the proposed mechanism.
[M.Nithya, Vijaykumar Varadarajan. **An Security Architecture for Campus Virtual Organization Using Role Based Access Control**. *Life Sci J* 2013;10(7s): 447-453] (ISSN:1097-8135). http://www.lifesciencesite.com.

Keywords: Virtual Organization (VO), Security, Role Based Access Control (RBAC), Community Authorization Service (CAS), Security Assertion Markup Language (SAML)

1. Introduction

Virtual communities/Virtual Organizations (VO) have many members participating as resource provider/consumer or both [1]. Expressing policies regarding direct trust relationships between producers and consumers has the problems of scalability, flexibility, expressibility and lack of policy hierarchies in such distributed environment. It is to offset these issues that a third party, a CAS server is introduced to manage policies governing access to a community's resources [2]. The CAS server tracks membership and users fine-grained access control policies. VO is provided a fine-grained mechanism by CAS to manage delegated policy spaces, allowing expression and enforcing expressive, consistent policies across resources and multiple independent policy domains. A user wanting to access community resources gets in touch with a CAS server that delegates user rights based on request and user's role within a community. The CAS architecture taps public keys enabling authentication/authorization mechanisms to address single sign on, delegation and credential mapping issues in VO settings [3].

Role Based Access Control (RBAC) approach enables users to be given VO roles memberships and role hierarchies [4]. CAS provides maintenance, granting users VO role memberships. Each organization's resource provider maintains only mapping information from VO roles in a database [3] thereby lowering the number of entries in the role-map file. Also, resource providers decide specific users access requests by maintaining role map file for authorization information thereby enabling resource providers with total authority over resources. VO's dynamic nature allows new users and resources to join VO. Existing users/resources can also either temporarily or permanently leave it without affecting resource providers as the CAS server looks after granting/revoking VO roles membership. New VOs trust relations between CAS server and VO can be direct/mutual or established via intermediate Trust Management Service.

RBAC with CAS produces a strong security combination. Assertion plays a crucial role to be implemented using Security Assertion Markup Language (SAML) [5] as VO is a heterogeneous and distributed environment. But RBAC resource utilization is higher than MAC. Lowering authentication time further improves VO performance.

Microsoft's threat modeling application was used to generate a broad spectrum threat model for campusGrid using current grid security mechanisms extensively with digital certificates [6, 7]. Its block diagram is seen in Figure 1.

Countermeasures suggested for the campus grid from the threat model include

- Using well known cryptographic algorithms
- Using cryptographically generated random keys
- Secure communication channel

Various researches drew good results in security following implementation of the recommended threat model [8, 9]. A major issue in using certificate-based

authentication is performance based, as public key (asymmetric) cryptography is at least 1000 times slower than secret key (symmetric) cryptography [10]. Maintaining certificate revocation lists which are to be stored on public servers requiring regular updating is another drawback.

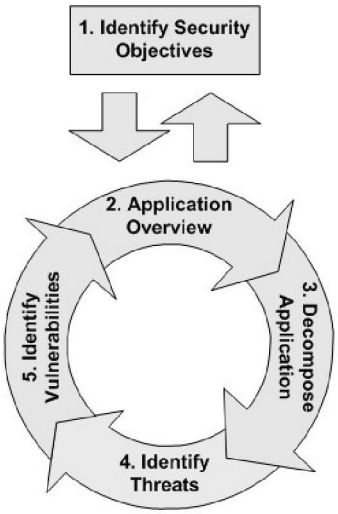

Figure 1: Threat model life cycle.

CAS server has entries for Certifying Authority (CA), users, servers and resources that include the community and groups organizing such entities. It also has policy statements which permit concern separation between site policies and VO policies. It provides a VO mechanism to manage delegated policy spaces, allowing expression and enforcing expressive, consistent policies across resources spanning many independent policy domains. VO is permitted by CAS to maintain its own set of policies. The sites combine local policies which are then enforced. The proposed architecture implements a local CAS (LCAS) managed within VO to reduce bandwidth overheads and simultaneously to use advantages implemented in CAS for Single Sign On (SSO).

2. Related Works

Geetha Kumari et al., [11] proposed a role based grid delegation model (RB-GDM) for grid security built over Role based access model (RBAC). Dynamic delegation, partial/ restricted delegation and coarse-grained/fine-grained delegation requirements can be realized through the proposed model. Intra and inter domain grid systems are provided a different framework for delegation requirements of peer to peer relationship and hierarchical role relationships.

Chadwick et al., [12] proposed PrivilEge and Role Management Infrastructure Standards (PERMIS) a role based access control mechanism using X.509 attribute certificates for storing users' roles. Authorization policy is written in XML, digitally signed and secured as an X.509 attribute certificate in PERMIS. The policy supports hierarchical RBAC, through which users are provided roles and roles/attributes are granted access rights. Superior roles/attributes inherit subordinate roles/attributes privileges. The creator stores the policy as a digitally signed attribute certificate in a Lightweight Directory Access Protocol (LDAP) directory. Attribute certificates conforming to the X.509 standard are user authorization tokens. LDAP directory servers store attribute certificates. Attribute acquisition is usually through a Privilege Allocator tool which creates X.509 attribute certificates, storing them in an LDAP server in the attribute certificate holder's entry. The subject might possess credentials from many different AAs that could be pre-issued, long lived and stored in a repository or short lived and issued on demand, according to Credential Issuing Policies.

Services oriented grids are more prominent than other grids in distributed environments. With the advent of online government services, governmental grids are expected to come up in huge numbers. Apart from security issues as in other grids, authorization in service oriented grids faces has shortcomings which need to be looked upon differently. Prasad et al [13] presented a CMMS model that overcomes all shortcomings adding to the simplicity of implementation due to its similarities with government services and functioning. CMMS suggests multilevel authorization, one each at VO and service levels. It also introduces the concept of states to users, VO and service nodes. Though it is suggestive of one way state-policy to service mapping, the two-way model is also similar and is framed easily. The proposed model requires some technological changes in PKI and X.509. Usually, states require monitoring through MONITOR nodes in CA; And X.509 certificates should contain state or policy or both. The model is a prototype of State Police Information Grid (SPIG). Only minor technological restructuring is required in PKIX and X.509 certificates.

3. Methodology

SAML defines an XML vocabulary to share security assertions [14]. SAML defines an XML-based framework to describe and exchange security information between identity and service providers and is also capable of providing secure single sign on solutions to internet service providers. Organization for the Advancement of Structured Information Standards (OASIS) defined SAML 2.0 standard in 2005. SAML 2.0 supports W3C XML encryption

satisfying privacy requirements. Another SAML 2.0 advantage is the support for service provider initiated by web single sign-on exchanges which allows a service provider to query an identity provider for authentication. Additionally, SAML 2.0 provides Single Logout functionality. SAML is built on existing web standards.

SAML assertion has security statements transferred between identity and service providers. Security information is expressed in portable SAML assertions which can be unsigned, signed or signed and encrypted depending on data type and application requirements sensitivity. SAML standard ensures message integrity by supporting X.509 digital signatures in the request/response transmissions. Again SAML supports and recommends HTTP over SSL 3.0 and TLS 1.0 in situations requiring data confidentiality. SAML assertions provide three statements.

- Authentication statements
- Attribute statements
- Authorization decision statements

Use of SAML assertions with WS-Security is described in the following steps (Figure 2):

A SOAP message sender obtains a SAML assertion by SAML Request/Response protocol/other methods. The following steps protect the SOAP message:

1. The sender constructs a SOAP message, including a SOAP header with a WS-Security header. Then a SAML assertion is placed within a WS-Security token. It is then included in the security header. A key constructs a digital signature over data in SOAP message body. The security header also includes signature information. This key is referred to by SAML assertion.
2. The message receiver verifies the digital signature.
3. The information in SAML assertion is used for Access Control and Audit logging

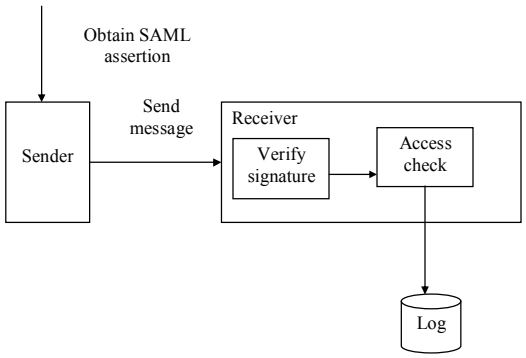

Figure 2: SAML assertion

SAML defines set of request/response protocols to communicate the assertions between identity and service providers. SAML bindings map SAML protocol message to standard network communication protocol message format to transport SAML assertions between identity and service providers. Profiles are the highest SAML component level defining how assertion, protocol and bindings combine to support use cases. SAML has many security mechanisms to detect/protect it from attacks, with the primary mechanism relying on a PKI.

Proposed Security Architecture – Low Resource Access Control (LRAC)

A disadvantage in use of public-key for authentication is additional computing overheads that increase the computational overheads [10]. There are many secret-key encryption methods which are faster than most public-key encryption methods now available. Another disadvantage is that a successful attack on a certification authority compromises the whole security mechanism allowing an adversary to impersonate through the use of a public-key certificate from the compromised authority, binding a key of the adversary's choice to another user's name. The proposed Low resource – Access Control (LRAC) comprises of

- A novel authentication mechanism using the public key concept being generated from local CAS and not CA to increase speed.
- Avoiding proxy CAS credentials by proposing a novel key and token mechanism generated dynamically to speed up the system.

This research proposes architecture that tries to access resources with keys and a token generated dynamically instead of signing in with credentials for every resource access by users. As only request ids are passed to VO it reduces chances of user credentials being revealed to the VO. The model is also designed that when a user tries to use the request id sent by another user to access resource, it is rejected after evaluation as request id for each user is generated dynamically for each request. So, the proposed architecture is simple, secure and faster than earlier approaches. Table 1 reveals a list of the security framework parameters where the master acts as local CAS server.

The concept of maintaining the user credentials and organizational details in a role based grid computing VO organization is explained step by step as follows and is shown in Figure 3.

Table 1: List of Parameters

Parameter	Description
p_k	Public Key
k_s	Secret Key
k_i	Intermediate Key
k_{vo}	VO Key
C_{id}	Confirmation ID
R_{id}	Request ID
V_T	Validation Token
V_{id}	VO ID
A_{id}	Aggregated ID
I_T	Intermediate Token

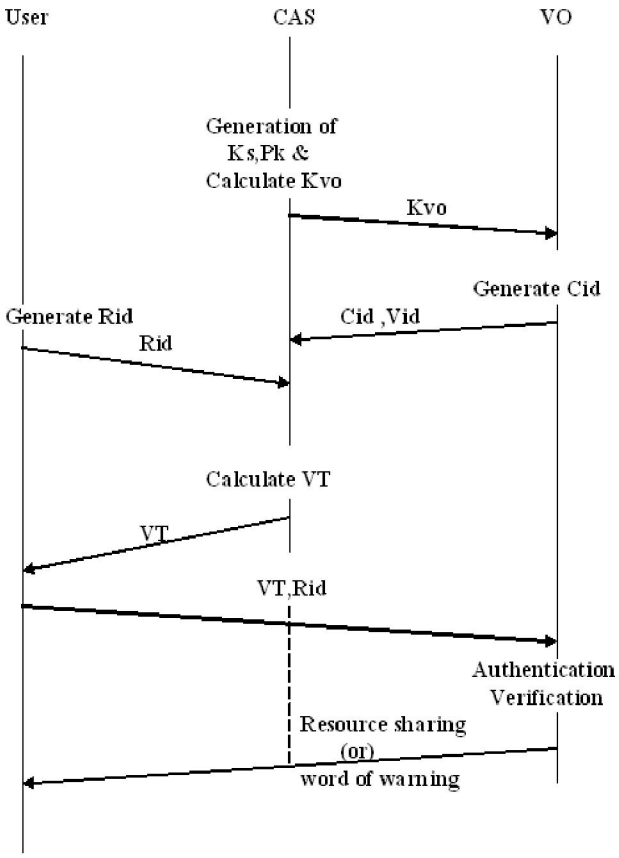

Figure 3: Flow diagram of the proposed architecture

Step 1: For every session or particular time period, local CAS generates a public key p_k and private key k_s pair using CPAN, open source software for key generation maintained in the CAS.

An intermediate key k_i is derived by

$$k_i = p_k^{k_s}$$

On this basis, a VO key named as k_{vo} is calculated by multiplying k_s with the arbitrary value to the power of k_s - 1.This is represented by the equation

$$k_{vo} = p_k^{k_s - 1} * k_s$$

CAS now sends k_{vo} to VO for further calculation that is to be performed later in this architecture. In the local CAS, the keys are maintained till the end of each session.

Step 2: VO stores k_{vo} as the VO key throughout the session for which a given job is executed. It then generates a confirmation id C_{id} in response to k_{vo} . VO now sends its own VO id V_{id} , an arbitrary real number and its corresponding C_{id} generated randomly to CAS to validate the user, so that CAS will hold V_{id} s for different VOs and its corresponding C_{id} s.

Step 3: When a user from an organization in the VO needs to access VO resources, it sends a Request id R_{id} which is nothing but an arbitrary integer to CAS using SAML for assertion.

Step 4: After receiving the user request in terms of R_{id} using SAML for the assertion, the CAS validates the received request id, R_{id} as genuine request based on SAML assertion. Once validation is over, CAS generates a validation token V_T for a respective user containing the role mentioned for the user by CAS. This is done by generating an intermediate token I_T given by

$$I_T = \left[1 + \frac{C_{id}}{R_{id}}\left[1 + \frac{k_i}{p_k}\right] - \frac{A_{id}}{R_{id}}\right]$$

Where $A_{id}=C_{id}+R_{id}$

This I_T is multiplied with the k_s by the admin module of CAS, and thus the final V_T is calculated.

The value of V_T is provided by

$$V_T = k_s * I_T$$

The admin sends this to the user directly as it will not reveal its private key to the proposed architecture module.

Step 5: The user receives the V_T from the CAS and sends it to VO along with the R_{id} generated in step 4.

Step 6: The Eventual Token validation process begins here. After receipt of the V_T and R_{id} from user, VO performs Token validation as follows

$$\log[V_T] - \log\left[\frac{C_{id}}{R_{id}} k_{vo}\right] = 0$$

When the above condition is satisfied, VO allows a user to access resources. Otherwise, a warning is issued to the adversary user.

The proposed authentication mechanism is implemented using SAML. The proposed CAS architecture was configured to utilize RBAC policies. The proposed algorithm for authentication was implemented instead of CA.

4. Results and Discussion

The VO consists of five colleges participating and sharing resources as shown in figure 4. It is seen from the figure that 4 colleges participate in the VO with resources shared as shown in table 2. The implementation was tested in the campus VO with 200 tasks being executed for each policy and with uniform communication and computation size.

Table 2: Resource snapshot provided by College.

	Number of Resources Provided
College 1	6
College 2	4
College 3	3
College 4	1

The Grid network operates in a Master/Slave mode with one node from college 1 acting as Resource allocator. All users were assigned 200 jobs with computation size of 5000000 and communication size of 100000. The experimental results of the proposed approach are presented here. The proposed approach is tested with valid users and adversary users set. First the user transmits a request id to CAS and the latter verifies this key with confirmation key from VO. CAS then provides Validation token which validates the user. Inputs from two different user classifications are evaluated. The result reveals user validation by verifying the validation code and thus identifying whether it is a bona fide user. A user is identified as invalid when validation fails.

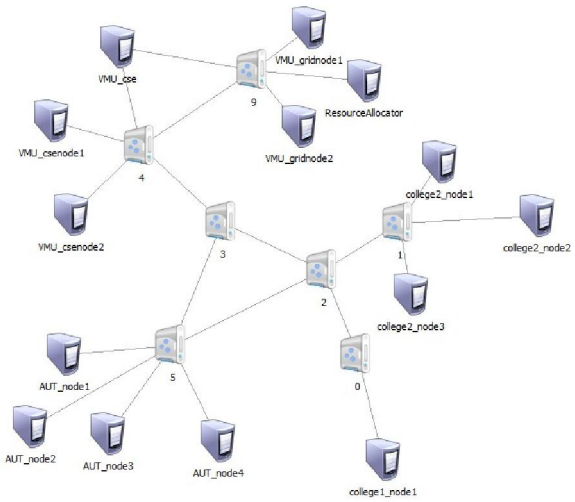

Figure 4: The proposed grid network

Inputs from two different user classifications are evaluated, and respective results at every stage are represented in a tabular column below. Some examples for valid users and invalid users are given in Table 4 and Table 5.

A user is identified to be invalid when the validation fails. Table 5 below, clearly shows that validation code is not balanced and, so a user is evaluated as invalid.

Authentication is a part of the proposed security model that includes authentication, authorization and assertion. As resources change dynamically and policies require administering along multiple resources, time for authentication plays a crucial part in the system's overall performance.

As this research focuses not only on security but also on faster access control mechanism, the proposed authentication mechanism in Low Resource Access Control (LRAC) is justified.

Table 3: Task completion time using proposed access control method

VO	Time to complete task using LRAC in Seconds
College1	12.672
College2	20.05
College3	14.968
College4	11.914

Table 4: Evaluation for Valid Users

Sl. No.	Arbitrary Value p_k	Secret Key k_s	Intermediate Key k_i	VO Key k_{vo}	Request Id R_{id}	Confirmation Id C_{id}	Aggregated Id A_{id}	$\log V_T$	$\log\left[\frac{C_{id}}{R_{id}} k_{vo}\right]$	Validation Is User Valid
1	2.0	9.0	512.0	2304.0	84.0	34.0	118.0	6.837946	6.837946	Valid User
2	1.0	145.0	1.0	145.0	11.0	13.0	24.0	5.143788	5.143788	Valid User
3	7.0	149.0	8.310132762606204E125	1.768871116611892E127	53.0	14.0	67.0	291.66742	291.66742	Valid User
4	1.0	237.0	1.0	237.0	70.0	87.0	157.0	5.685473	5.685473	Valid User
5	6.0	66.0		2.5082753518537416E52	74.0	25.0	99.0	119.56883	119.56883	Valid User

Table 5: Evaluation for Invalid Users

Arbitrary Value p_k	Secret Key k_s	Intermediate Key k_i	VO Key k_{vo}	Request Id R_{id}	Confirmation Id C_{id}	Validation Id A_{id}	$\log V_T$	$\log\left[\frac{C_{id}}{R_{id}} k_{vo}\right]$	Validation Is User Valid
1.0	151.0	1.0	151.0	12.0	40.0	52.0	3.8233573	6.2212524	Invalid User
7.0	35.0	3.788186922656648E29	1.8940934613283236E30	66.0	87.0	153.0	69.85147	69.992546	Invalid User

Grid performance improved considerably when compared to Role Based Access Control mechanism. Grid performance comparison under 3 categories is seen in figure 5.

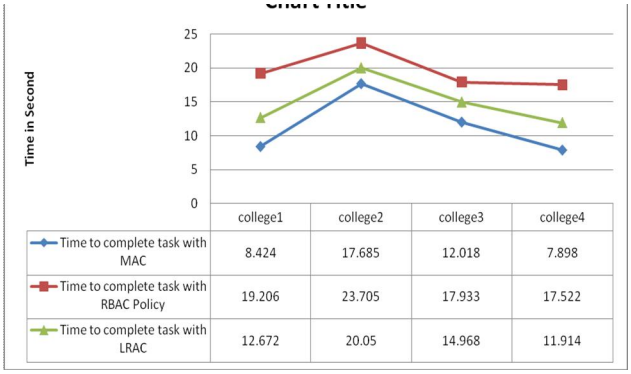

Figure 5: Time taken by different mechanisms to compute the same set of tasks.

Figure 5 shows that VO performance improves with the proposed mechanism which is linear under different policies revealing that the proposed mechanism can be scaled for large networks.

5. Conclusion

As Organizations included in a VO are dynamic and with the VO itself being created in a dynamic manner, it could lead to threats of user credentials being exploited illegally. An adversary user who has already hacked user credential can easily misuse VO resources with known credential revealed to VOs. Hence, such systems lack security. The proposed Low resource – Access Control (LRAC) is a novel authentication mechanism which uses the concept of public key being generated from the local CAS and not CA to increase speed. It also avoids CAS used proxy credentials by proposing a novel key and token mechanism generated dynamically and which thereby speeds the system up. Simulation results reveal that grid performance improved greatly when compared to Role Based Access Control mechanism.

References

1. Deitos, R., Kerschbaum, F., & Robinson, P. (2006, November). A comprehensive security architecture for dynamic, web service based virtual organizations for businesses. In Proceedings of the 3rd ACM workshop on Secure web services (pp. 103-104). ACM.
2. Pearlman L., Welch V., Foster I., Kesselman C. and Tuecke S. (2002), 'A Community Authorization Service for Group Collaboration'. Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks.
3. Anil Pereira L., Vineela Muppavarapu and Soon Chung M. (2006), 'Role-Based Access Control

for Grid Database Services Using the Community Authorization Service', IEEE Transactions on Dependable and Secure Computing, Vol. 3, No. 2, pp. 156-166.
4. Muppavarapu, V., Pereira, A. L., & Chung, S. M. (2010). Role-based access control for a Grid system using OGSA-DAI and Shibboleth. The Journal of Supercomputing, 54(2), 154-179.
5. Riedel, M., Streit, A., Lippert, T., Wolf, D. K. F., & Kranzlmueller, D. (2009, September). Concepts and design of an interoperability reference model for scientific-and grid computing infrastructures. In Proceedings of the Applied Computing Conference, Athens, Greece.
6. Alfieri R., Cecchini R., Ciaschini V., dell'Agnello L., Gianoli A. and Spataro F. (2003), 'VOMS: an Authorization System for Virtual Organizations', Presented at the 1st European across Grids Conference, Santiago de Compostela, pp. 13-14.
7. Metke, A. R., & Ekl, R. L. (2010). Security technology for smart grid networks. Smart Grid, IEEE Transactions on, 1(1), 99-107.
8. Lepro R. (2003), 'Cardea: Dynamic Access Control in Distributed Systems', NASA Technical Report NAS-03-020.
9. Cantor S., Kemp J., Maler E. and Philpott R. (2005), 'Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.02', OASIS Standard Specification.
10. Bela, G., & Piroska, H. (2009). A Chained Authentication Model for Virtual Organizations.
11. Geethakumari, G., Negi, D. A., & Sastry, D. V. (2008). RB-GDM: A Role-Based Grid Delegation Model. International Journal of Computer Science and Security, 2(1), 61.
12. Chadwick, D. W., & Otenko, A. (2003). The PERMIS X. 509 role based privilege management infrastructure. Future Generation Computer Systems, 19(2), 277-289.
13. Prasad, A., Verma, S. S., & Sharma, A. K. (2010). Certification Authority Monitored Multilevel and Stateful Policy Based Authorization in Services Oriented Grids. arXiv preprint arXiv:1007.0295.
14. Cantor, S., Hirsch, F., Kemp, J., Philpott, R., & Maler, E. (2005). Bindings for the OASIS Security Assertion Markup Language (SAML) V2. 0.

1/8/2013